NAACL-HLT 2021

Advances in Language and Vision Research

Proceedings of the Second Workshop

June 11, 2021

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-954085-37-4 (Volume 1)

Introduction

Language and vision research has attracted great attention from both natural language processing (NLP) and computer vision (CV) researchers. Gradually, this area is shifting from passive perception, templated language and synthetic imagery or environments to active perception, natural language and real-world environments. Thus far, few workshops on language and vision research have been organized by groups from the NLP community. This year, we are organizing the second workshop on Advances in Language and Vision Research (ALVR) in order to promote the frontier of language and vision research and bring interested researchers together to discuss how to best tackle real-world problems in this area.

This workshop covers (but is not limited to) the following topics:

- New tasks and datasets that provide real-world solutions in the intersection of NLP and CV;
- Language-guided interaction with the real world, e.g. navigation via instruction following or dialogue;
- External knowledge integration in visual and language understanding;
- Visually grounded multilingual study, e.g. multimodal machine translation;
- Fairness in multimodal machine learning;
- Shortcoming of existing language and vision tasks and datasets;
- Benefits of using multimodal learning in downstream NLP tasks;
- Self-supervised representation learning in language and vision;
- Transfer learning (including few/zero-shot learning) and domain adaptation;
- Cross-modal learning beyond image understanding, such as videos and audios;
- Multidisciplinary study that may involve linguistics, cognitive science, robotics, etc.

The details of our workshop can be found at `https://alvr-workshop.github.io/`.

Proceedings of the ALVR workshop from previous years can be found on ACL Anthology:
`https://www.aclweb.org/anthology/venues/alvr/`

Organizers:

Xin (Eric) Wang, UC Santa Cruz
Ronghang Hu, Facebook AI Research
Drew Hudson, Stanford
Tsu-Jui Fu, UC Santa Barbara
Marcus Rohrbach, Facebook AI Research
Daniel Fried, UC Berkeley

Program Committee:

Shubham Agarwal, Heriot-Watt University
Arjun Akula, University of California, Los Angeles
Asma Ben Abacha, NIH/NLM
Luciana Benotti, The National University of Cordoba
Khyathi Raghavi Chandu, Carnegie Mellon University
Angel Chang, Stanford University
Dhivya Chinnappa, Thomson Reuters
Abhishek Das, Facebook AI
Simon Dobnik, University of Gothenburg
Thoudam Doren Singh, National Institute of Technology, Silchar, India
Hamed Firooz, Facebook AI
Zhe Gan, Microsoft
Cristina Garbacea, University of Michigan
Jack Hessel, AI2
Gabriel Ilharco, University of Washington
Shailza Jolly, TU Kaiserslautern Germany
Marimuthu Kalimuthu, Saarland University, Saarland Informatics Campus
Noriyuki Kojima, Cornell University
Christopher Kummel, Beuth University of Applied Sciences Berlin
Loitongbam Sanayai Meetei, National Institute of Technology Silchar, India
Khanh Nguyen, University of Maryland
Yulei Niu, Renmin University of China
Aishwarya Padmakumar, University of Texas, Austin
Hamid Palangi, Microsoft Research
Shruti Palaskar, Carnegie Mellon University
Vikas Raunak, Carnegie Mellon University
Arka Sadhu, University of Southern California
Alok Singh, National Institute of Technology, Silchar India
Alane Suhr, Cornell University
Hao Tan, University of North Carolina
Xiangru Tang, University of the Chinese Academy of Sciences, China
Ece Takmaz, University of Amsterdam

Invited Speaker:

Jacob Andreas, MIT
Jason Baldridge, Google
Mohit Bansal, UNC Chapel Hill
Yonatan Bisk, Carnegie Mellon University
Joyce Y. Chai, University of Michigan
Yejin Choi, University of Washington

Raymond J. Mooney, University of Texas at Austin
Anna Rohrbach, UC Berkeley
Kate Saenko, Boston University
William Wang, UC Santa Barbara

Table of Contents

Feature-level Incongruence Reduction for Multimodal Translation

Zhifeng Li and **Yu Hong**✉ and **Yuchen Pan** and **Jianmin Yao** and **Guodong Zhou**
Institute of Artificial Intelligence, Soochow University
School of Computer Science and Technology, Soochow University
No.1, Shizi ST, Suzhou, China, 215006
{lizhifeng0915,tianxianer,yuchenpan59419}@gmail.com
johnnytang1120@gmail.com, {gdzhou,jyao}@suda.edu.cn

Abstract

Caption translation aims to translate image annotations (captions for short). Recently, Multimodal Neural Machine Translation (MNMT) has been explored as the essential solution. Besides of linguistic features in captions, MNMT allows visual (image) featuresto be used. The integration of multimodal features reinforces the semantic representation and considerably improves translation performance. However, MNMT suffers from the incongruence between visual and linguistic features. To overcome the problem, we propose to extend MNMT architecture with a harmonization network, which harmonizes multimodal features (linguistic and visual features) by unidirectional modal space conversion. It enables multimodal translation to be carried out in a seemingly monomodal translation pipeline. We experiment on the golden Multi30k-16 and 17. Experimental results show that, compared to the baseline, the proposed method yields the improvements of 2.2% BLEU for the scenario of translating English captions into German (En→De) at best, 7.6% for the case of English-to-French translation (En→Fr) and 1.5% for English-to-Czech (En→Cz). The utilization of harmonization network leads to the competitive performance to the-state-of-the-art.

1 Introduction

Caption translation is required to translate a source-language caption into target-language, where a caption refers to the sentence-level text annotation of an image. As defined in the shared multimodal translation task[1] in WMT, caption translation can be conducted over both visual features in images and linguistic features of the accompanying captions. The question of how to opportunely utilize images for caption translation motivates the study of multimodality, including not only the extraction of visual features but the cooperation between visual and linguistic features. In this paper, we follow the previous work (Specia et al., 2016) to boil caption translation down to a problem of multimodal machine translation.

So far, a large majority of previous studies tend to develop a neural network based multimodal machine translation model (viz., MNMT), which consists of three basic components:

- **Image encoder** which characterizes a captioned image as a vector of global or multi-regional *visual features* using a convolutional neural network (CNN) (Huang et al., 2016).
- **Neural translation network** (Caglayan et al., 2016; Sutskever et al., 2014; Bahdanau et al., 2014) which serves both to encode a source-language caption and to generate the target-language caption by decoding, where the latent information that flows through the network is referred to *linguistic feature*.
- **Multimodal learning network** which uses visual features to enhance the encoding of linguistic semantics (Ngiam et al., 2011). Besides of the concatenation and combination of linguistic and visual features, vision-to-language attention mechanisms serve as the essential operations for cross-modality learning. Nowadays, they are implemented with single-layer attentive (Caglayan et al., 2017a; Calixto et al., 2017b), doubly-attentive (Calixto et al., 2017a), interpolated (Hitschler et al., 2016) and multi-task (Zhou et al., 2018) neural networks, respectively.

Multimodal learning networks have been successfully grounded with different parts of various neural translation networks. They are proven effective in enhancing translation performance. Nevertheless, the networks suffer from incongruence between visual and linguistic features because:

- Visual and linguistic features are projected into incompatible semantic spaces and therefore fail to be corresponded to each other.

[1] http://www.statmt.org/wmt16/

Proceedings of the Second Workshop on Advances in Language and Vision Research, pages 1–10

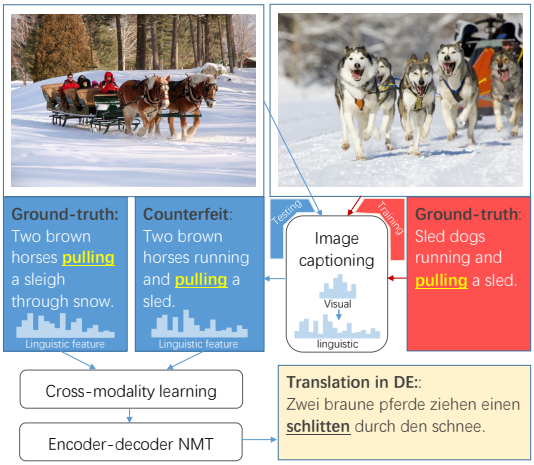


Figure 1: An example in which image captioning contributes to the reduction of incongruence.

- Linguistic features are sequence-dependent. This is attributable to pragmatics, syntax or even rhetoric. On the contrary, visual features are sequence-independent but position-sensitive. This is attributable to spatial relationships of visual elements. Thus, a limited number of visual features can be directly used to improve the understanding of linguistic features and translation.

Considering the Figure 1("*Counterfeit*" means Image Captioning output), the visual features enable a image processing model to recognize "*two horses*" as well as their position relative to a "*sleigh*". However, such features are obscure for a translation model and useful for translating a verb, such as "*pulling*" in the caption. In this case, incongruence of heterogeneous features results from the unawareness of the correspondence between spatial relationship ("*running horses*" ahead of "*sleigh*") and linguistic semantics ("*pulling*").

To ease the incongruence, we propose to equip the current MNMT with a harmonization network, in which visual features are not directly introduced into the encoding of linguistic semantics. Instead, they are transformed into linguistic features before absorbed into semantic representations. In other words, we tend to make a detour during the cross-modality understanding, so as to bypass the modality barrier (Figure 2). In our experiments, we employ a captioning model to conduct harmonization. The hidden states it produced for decoding caption words are intercepted and involved into the representation learning process of MNMT.

The rest of the paper is organized as follows: Section 2 presents the motivation and methodological framework. Section 3 gives the NMT model we use. In Section 4, we introduce the captioning model that is trainable for cross-modality feature space transformation. Section 5 presents the captioning based harmonization networks as well as the resultant MNMT models. We discuss test results in Section 6 and overview the related work in Section 7. We conclude the paper in section 8.

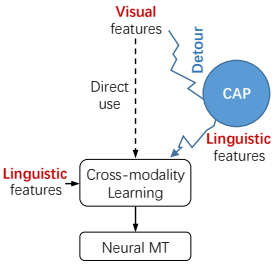


Figure 2: Bypassing modality barrier by captioning.

2 Fundamentals and Methodological Framework

We utilize Anderson et al (2018)'s image captioning (CAP for short) to guide the cross-modality feature transformation, converting visual features into linguistic. CAP is one of the generation models which are specially trained to generate language conditioned on visual features of images. Ideally, during training, it learns to perceive the correspondence between visual and linguistic features, such as that between the spatial relationship of "*running dogs ahead of a sled*" in Figure 1 and the meaning of the verb "*pulling*". This allows CAP to produce appropriate linguistic features during testing in terms of similar visual features, such as that in the case of predicting the verb "*pulling*" for the scenario of "*running horses ahead of a sleigh*".

Methodologically speaking, we adopt the linguistic features produced by the encoder of CAP instead of the captions generated by the decoder of CAP. On the basis, we integrate both the linguistic features of the original source-language caption and those produced by CAP into Calixto et al (2017b)'s attention-based cross-modality learning model (see Figure 3). Experimenal results show that the learning model substantially improves Bahdanau et al (2014)'s encoder-decoder NMT system.

3 Preliminary 1: Attentive Encoder-Decoder NMT (Baseline)

We take Bahdanau et al. (2014)'s attentive encoder-decoder NMT as the baseline. It is constructed

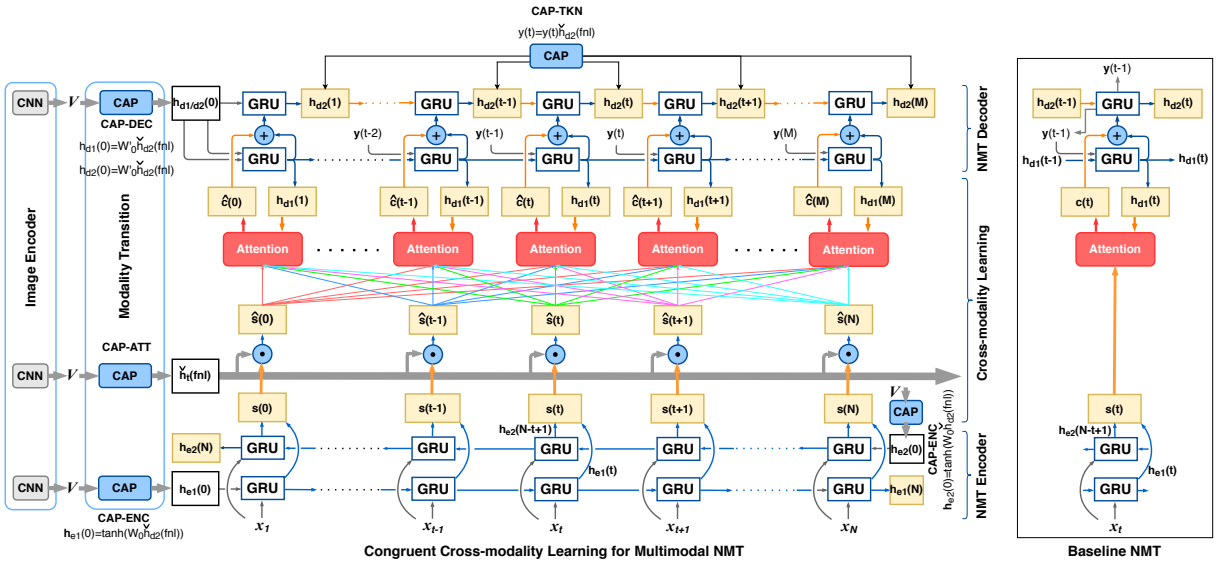


Figure 3: The overall architecture of MNMT.

with a BiGRU encoder and a Conditional GRU (CGRU) decoder (Firat and Cho, 2016; Caglayan et al., 2017a). Attention mechanism is used between BiGRU and CGRU. The diagram at the right side of Figure 3 shows the baseline framework.

For a source-language caption, we represent it with a sequence of randomly-initialized (Kalchbrenner and Blunsom, 2013) word embeddings X=$(x_1, ..., x_N)$, where each x_t is uniformly specified as a k-dimensional word embedding. Conditioned on the embeddings, Chung et al (2014)'s BiGRU is used to compute the bidirectional hidden states S=$(s_1, ..., s_N)$, where each s_t is obtained by combining the t-th hidden state of forward GRU and that of backward GRU: s_t=$[\overrightarrow{GRU}^e(x_t), \overleftarrow{GRU}^e(x_t)]$. Padding (Libovický and Helcl, 2018) and dynamic stabilization (Ba et al., 2016) are used.

Firat and Cho (2016)'s CGRU is utilized for decoding, which comprises two forward GRU units, i.e., $\overrightarrow{GRU}^{d1}$ and $\overrightarrow{GRU}^{d2}$ respectively. $\overrightarrow{GRU}^{d1}$ plays the role of producing the inattentive decoder hidden states H^{d1}=$(h_1^{d1}, ..., h_M^{d1})$, where each h_t^{d1} is computed based on the output state h_{t-1}^{d1} and prediction y_{t-1} at the previous time step: h_t^{d1}=$\overrightarrow{GRU}^{d1}(h_{t-1}^{d1}, y_{t-1})$ (Note: the prediction y_t denotes the k-dimensional embedding of the predicted word at the t-th decoding step). By contrast, $\overrightarrow{GRU}^{d2}$ serves to produce the attentive decoder hidden states H^{d2}=$(h_1^{d2}, ..., h_M^{d2})$, where each h_t^{d2} is computed conditioned on the previous attentive state h_{t-1}^{d2}, the current inattentive state h_t^{d1}, as well as the current attention-aware context c_t: h_t^{d2}=$\overrightarrow{GRU}^{d2}(h_{t-1}^{d2}, h_t^{d1} \oplus c_t)$. The context c_t is obtained by the attention mechanism over the global encoder hidden states S: $c_t = \alpha_t S$, where α_t denotes the attention weight at the t-th time step. Eventually, the prediction of each target-language word is carried out as follows (where, W_h, W_c, W_y, b_o and b_y are trainable parameters):

$$\mathcal{D}_t(y_{t-1}, h_t^{d2}, c_t) \sim \begin{cases} o_t = tanh(y_{t-1} + \mathrm{W}_h h_t^{d2} + \\ \quad \mathrm{W}_c c_t + b_o) \\ P(y_t|o_t) = \mathrm{softmax}(\mathrm{W}_y^\top o_t \\ \quad + b_y) \end{cases} \quad (1)$$

4 Preliminary 2: Image-dependent Linguistic Feature Acquisition by CAP

For an image, captioning models serve to generate a sequence of natural language (caption) that describes the image. Such kind of models are capable of transforming visual features into linguistic features by encoder-decoder networks. We utilize Anderson et al. (2018)'s CAP to obtain the transformed linguistic features.

4.1 CNN based Image Encoder

What we feed into CAP is a full-size image which needs to be convolutionally encoded beforehand. He et al (He et al., 2016a)'s CNNs (known as ResNet) with deep residual learning mechanism (He et al., 2016b) is capable of encoding images. In our experiments, we employ the recent version

of ResNet, i.e., ResNet-101 , which is constructed with 101 convolutional layers. It is pretrained on ImageNet (Russakovsky et al., 2015) in the scenario of 1000-class image classification.

Using ResNet-101, we characterize an image as a convolutional feature matrix: $V \in \mathbb{R}^{k\times 2048} = \{v_1, ..., v_k\}$, in which each element $v_i \in \mathbb{R}^{2048}$ is a real-valued vector and corresponds to an image region in the size of 14×14 pixels.

4.2 Top-down Attention-based CAP

CAP learns to generate a caption over V. It is constructed with two-layer RNNs with LSTM (Anderson et al., 2018), LSTM1 and LSTM2 respectively. LSTM1 (in layer-1) computes the current first-layer hidden state $\check{h}_t^{d1}$ conditioned on the current first-layer input $\check{x}_t^{d1}$ and previous hidden state $\check{h}_{t-1}^{d1}$: $\check{h}_t^{d1}$=LSTM1($\check{x}_t^{d1}$, $\check{h}_{t-1}^{d1}$). The input $\check{x}_t^{d1}$ is obtained by concatenating the previous hidden state $\check{h}_{t-1}^{d1}$ and previous prediction $\check{y}_{t-1}$, as well as the condensed global visual feature $\bar{v}$: $\check{x}_t^{d1}$=[$\bar{v}$, $\check{h}_{t-1}^{d1}$, $\check{y}_{t-1}$], where $\bar{v}$ is calculated by the normalized accumulation of overall convolutional features in V: $\bar{v} = \frac{1}{k}\sum_i v_i$ ($\forall v_i \in V$). We specify the first-layer hidden state as the initial image-dependent linguistic features.

Attention mechanism (Sennrich et al., 2015) is used for highlighting the attention-worthy image context, so as to produce the attention-aware vector of image context $\check{v}_t$: $\check{v}_t = \sum \check{\alpha}_t V$. The attention weight $\check{\alpha}_t$ is obtained by aligning the current image-dependent hidden state $\check{h}_t^{d1}$ with every convoluted visual feature v_i: $\check{\alpha}_t = \text{softmax}.f(\check{h}_t^{d1}, v_i)$, where $f(*)$ is the non-linear activation function.

LSTM2 (in layer-2) serves as a neural language model (viz., language-oriented generation model). It learns to encode the current second-layer hidden state $\check{h}_t^{d2}$ conditioned on the current second-layer input $\check{x}_t^{d2}$ and previous hidden state $\check{h}_{t-1}^{d2}$: $\check{h}_t^{d2}$=LSTM1($\check{x}_t^{d2}$, $\check{h}_{t-1}^{d2}$). The input $\check{x}_t^{d2}$ is obtained by concatenating the current first-layer hidden state $\check{h}_t^{d1}$ (emitted from layer-1) and current attention-aware image context $\check{v}_t$: $\check{x}_t^{d2}$=[$\check{v}_t$, $\check{h}_t^{d1}$]. We specific a second-layer hidden state $\check{h}_t^{d2}$ as the image-dependent attention-aware linguistic features. Towards the image captioning task, CAP generally decodes the second-layer hidden states $\check{h}_t^{d2}$ to predict caption words. In our case, we tend to integrate them into multimodal NMT by cross-modality learning (see the next section).

5 Harmonization for MNMT

In the previous work of multimodal NMT, visual features in V are directly used for cross-modality learning. By contrast, we transform visual features into image-dependent attention-aware linguistic features (i.e., second-layer hidden states $\check{h}_t^{d2}$ emitted by CAP) before use. We provide four-class variants of cross-modality learning to improve NMT. They absorb image-dependent attention-aware linguistic features in different ways, including a variant that comprises attentive feature fusion (CAP-ATT) and three variants (CAP-ENC, CAP-DEC and CAP-TKN) which carry out reinitialization and target-language embedding modulation. Figure 3 shows the positions in the baseline NMT where the variants come into play.

CAP-ATT intends to improve NMT by conducting joint representation learning across the features of the source-language caption and that of the accompanying image. On one side, CAP-ATT adopts the encoder hidden state s_t (emitted by the BiGRU encoder of the baseline NMT) and uses it as the language-dependent linguistic feature. On the other side, it takes the image-dependent attention-aware linguistic feature $\check{h}_t^{d2}$ (produced by CAP). We suppose that the two kinds of features (i.e., $\check{h}_t^{d2}$ and s_t) are congruent with each other. On the basis, CAP-ATT blends $\check{h}_t^{d2}$ into s_t to form the joint representation $\widehat{s}_t$. Element-wise feature fusion (Cao and Xiong, 2018) is used to compute $\widehat{s}_t$: $\widehat{s}_t = s_t \odot \check{h}_t^{d2}$. Using the joint representation $\widehat{s}_t$, CAP-ATT updates the attention-aware context c_t which is fed into the CGRU decoder of the baseline NMT: $\widehat{c}_t$=$\alpha_t\widehat{S}$, $\forall\widehat{s} \in \widehat{S}$. By substituting the updated context $\widehat{c}_t$ into the computation of the CGRU decoder, CAP-ATT further refines the decoder hidden state $h_t^{d_2}$ and prediction of target-language words. Equation 2 formulates the decoding process, where $\mathcal{D}_t$ is the shorthand of equation (1).

$$\widehat{\mathcal{D}}_t(y_{t-1}, \widehat{h}_t^{d2}, \widehat{c}_t) \sim \begin{cases} \widehat{h}_t^{d_2} = \overrightarrow{GRU}^{d2}(\widehat{h}_{t-1}^{d_2}, h_t^{d_1} \\ \qquad \oplus \widehat{c}_t) \\ y_t \Leftarrow \mathcal{D}_t(y_{t-1}, \widehat{h}_t^{d_2}, c_t) \end{cases} \tag{2}$$

CAP-ENC reinitializes the BiGRU encoder of the baseline NMT with the final image-dependent attention-aware linguistic feature $\check{h}_t^{d2}$ (t=N) (produced by CAP): $\overleftarrow{h}_0 = \overrightarrow{h}_0 = \tanh(\mathrm{W}_0\check{h}_t^{d2})$, where $\overleftarrow{h}_0$ and $\overrightarrow{h}_0$ are the initial states of BiGRU, and W_0 refers to the trainable parameter. **CAP-**

DEC uses $\check{h}_t^{d2}$ (t=N) to reinitialize the CGRU decoder of the baseline NMT: $h_0^{d1} = h_0^{d2} = \tanh(\mathrm{W}_0'\check{h}_t^{d2})$, where h_0^{d1} and h_0^{d2} are the initial decoder hidden states of CGRU. Using $\check{h}_t^{d2}$ (t=N), **CAP-TKN** modulates the predicted target-language word embedding y_t at each decoding step: $y_t = y_t \odot \tanh(\mathrm{W}_{tkn}\check{h}_t^{d2})$, where W_{tkn} is the trainable parameter. **CAP-ALL** equips a MNMT system with all the variants.

6 Experimentation

6.1 Resource and Experimental Datasets

We perform experiments on Multi30k-16 and Multi30k-17[2], which are provided by WMT for the shared tasks of multilingual captioning and multimodal MT (Elliott et al., 2016). The corpora are used as the extended versions of Flichr30k (Young et al., 2014), since they contain not only English (En) image captions but their translations in German (De), French (Fr) and Czech (Cz). Hereinafter, we specify an example in Multi30k as an image which is accompanied by three En→De, En→Fr and En→Cz caption-translation pairs. Each of Multi30k-*16* and Multi30k-*17* contains about 31K examples. We experiment on the corpora separately, and as usual divide each of them into training, validation and test sets, at the scale of 29K, 1,014 and 1K examples, respectively.

In addition, we carry out a complementary experiment on the ambiguous COCO which contains 461 examples (Elliott et al., 2017). Due to the inclusion of ambiguous verbs, the examples in ambiguous COCO can be used for the evaluation of visual sense disambiguation in a MNMT scenario.

6.2 Training and Hyperparameter Settings

For preprocessing, we apply Byte-Pair Encoding (BPE) (Sennrich et al., 2015) for tokenizing all the captions and translations in Multi30k and COCO, and use the open-source toolkit[3] of Moses (Koehn et al., 2007) for lowercasing and punctuation normalization. It reproduces the neural network architecture of Anderson et al (Anderson et al., 2018)'s top-down attentive CAP. The only difference is that it merely utilizes ResNet-101 in generating the input set of visual features V, without the use of Faster R-CNN (Ren et al., 2015). This CAP has been trained on MSCOCO captions dataset (Lin et al., 2014) using the same hyperparameter settings as that in Anderson et al. (2018)'s work.

Besides of the baseline NMT (Bahdanau et al., 2014) mentioned in section 2, we compare our model with Caglayan et al (Caglayan et al., 2017a)'s convolutional visualfeature based MNMT. In this paper, we follow Caglayan et al (Caglayan et al., 2017a)'s practice to implement and train our model. First of all, we implement our model with the nmtpy framework (Caglayan et al., 2017b) using Theano v0.9. During training, ADAM with a learning rate of 4e-4 is used and the batch size is set as 32. We initialize all the parameters (i.e., transformation matrices and biases) using Xavier and clip the total gradient norm to 5. We drop out the input embeddings, hidden states and output states with the probabilities of (0.3, 0.5, 0.5) for En→De MT, (0.2, 0.4, 0.4) for En→Fr and (0.1, 0.3, 0.3) for En→Cz. In order to avoid overfitting, we apply a L_2 regularization term with a factor of $1e$-5. We specify the dimension as 128 for all token embeddings ($k = 128$) and 256 for hidden states.

6.3 Comparison to the Baseline

We carry out 5 independent experiments (5 runs) for each of the proposed MNMT variants. In each run, any of the variants is retrained and redeveloped under cold-start conditions using a set of randomly-selected seeds by MultEval[4]. Eventually, the resultant models are evaluated on the test set with BLEU (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2014) and TER (Snover et al., 2006).

For each variant, we report not only the comprehensive performance (denoted as $ensemble$) which is obtained using ensemble learning (Garmash and Monz, 2016) but that without ensemble learning. In the latter case, the average performance (μ) and deviations (σ) in the 5 runs are reported.

6.3.1 Performance on Multi30k

Tables 1 and 2 respectively show the performance of our models on Multi30k-16 and Multi30k-17 for the translation scenarios of En→De, En→Fr and En→Cz. Each of our MNMT models in the tables is denoted with a symbol "+", which indicates that a MNMT model is constructed with the baseline and one of our cross-modality learning models. The baseline is specified as the monomodal NMT model which is developed by Bahdanau et al. (2014) (as

[2]https://github.com/multi30k/dataset/tree/master/data-/task1/raw

[3]https://github.com/moses-smt/mosesdecoder/tree/-master/scripts/tokenizer

[4]https://github.com/jhclark/multeval

En→De	Multi30k-*16* ($\mu \pm \sigma$/ensemble)			Multi30k-*17* ($\mu \pm \sigma$/ensemble)		
	BLEU	METEOR	TER	BLEU	METEOR	TER
Baseline	38.1±0.8/40.7	57.3±0.5/59.2	N/A	30.8±1.0/33.7	51.6±0.5/53.8	N/A
+CAP−ATT	39.2±0.8/41.3	57.5±0.6/59.4	40.9±0.8/39.5	32.1±0.9/33.6	51.0±0.7/52.9	48.7±0.8/47.3
+CAP−ENC	39.1±0.8/41.2	57.6±0.7/59.2	40.9±0.8/39.3	32.5±0.8/33.8	**52.2±0.7/54.5**	**48.5±0.8/46.3**
+CAP−DEC	38.9±0.8/41.0	57.4±0.7/59.3	**41.3±0.8/39.1**	**33.0±0.8/34.3**	51.6±0.7/53.2	48.6±0.8/47.1
+CAP−TKN	39.1±0.8/40.9	57.3±0.6/58.6	**41.3±0.8/39.1**	32.2±0.8/33.9	51.3±0.7/53.5	48.5±0.8/47.0
+CAP−ALL	**39.6±0.9/42.1**	**57.5±0.7/59.9**	41.1±0.8/39.4	31.6±0.8/33.9	51.6±0.7/53.7	49.7±0.7/47.1
En→Fr	Multi30k-*16* ($\mu \pm \sigma$/ensemble)			Multi30k-*17* ($\mu \pm \sigma$/ensemble)		
	BLEU	METEOR	TER	BLEU	METEOR	TER
Baseline	52.5±0.3/54.3	69.6±0.1/71.3	N/A	50.4±0.9/53.0	67.5±0.7/69.8	N/A
+CAP−ATT	**60.1±0.8/63.3**	**74.3±0.6/77.1**	**25.1±0.7/22.7**	52.5±0.9/56.1	68.2±0.7/71.2	31.5±0.7/28.4
+CAP−ENC	59.3±0.9/62.8	73.5±0.6/76.4	26.2±0.7/23.3	52.2±0.8/55.8	68.1±0.7/71.1	31.5±0.7/28.5
+CAP−DEC	60.1±0.9/62.6	74.2±0.7/76.3	25.6±0.6/23.0	51.9±0.9/55.7	67.6±0.7/71.3	**31.6±0.7/28.1**
+CAP−TKN	60.3±0.8/63.0	74.5±0.6/76.6	25.2±0.6/23.0	52.7±0.9/56.0	**68.3±0.6/71.3**	31.5±0.7/28.6
+CAP−ALL	60.1±0.8/62.7	74.3±0.6/76.4	25.0±0.4/23.1	**52.8±0.9/56.1**	68.6±0.6/71.1	31.2±0.7/28.9

Table 1: Performance for both En→De and En→Fr on Multi30k-16 and Multi30k-17.

En→Cz	Multi30k(2016) ($\mu \pm \sigma$ /ensemble)		
	BLEU	METEOR	TER
Baseline	30.5±0.8/32.6	29.3±0.4/31.4	N/A
+CAP−ATT	31.8±0.9/33.4	**30.2±0.4/32.6**	46.1±0.8/43.6
+CAP−ENC	31.7±0.8/33.3	29.9±0.4/32.1	46.3±0.8/43.5
+CAP−DEC	31.6±0.9/33.3	30.0±0.4/32.3	45.6±0.8/43.6
+CAP−TKN	**32.0±0.9/33.9**	30.1±0.4/32.3	45.7±0.8/43.3
+CAP−ALL	31.8±0.9/33.6	29.9±0.4/31.5	**45.3±0.8/43.3**

Table 2: Performance for En→Cz on Multi30k-16

mentioned in section 2) and redeveloped as the baselines in a variety of research studies on multimodal NMT (Calixto et al., 2017a,b; Caglayan et al., 2017a). We quote the results reported in Caglayan et al. (2017a)'s work as they were better.

It can be observed that our MNMT models outperform the baseline. They benefits from the performance gains yielded by the variants of CAP based cross-modality learning, which are no less than 1.5% BLEU when ensemble learning is used, and 0.6% when not to use it. In particular, +CAP-ATT obtains a performance increase of up to 7.6% BLEU (μ) in the scenario of En→Fr MT. The gains in METEOR score we obtain are less obvious than that in BLEU, which is about 5.3% (μ) at best.

We follow Clark et al. (2011) to perform significance test. The test results show that +CAP-ATT, +CAP-DEC and +CAP-TKN achieve a p-value of 0.02, 0.01 and 0.007, respectively. Clark et al. (2011) have proven that the performance improvements are significant only if the p-value is less than 0.05. Therefore, the proposed method yields statistically significant performance gains.

6.3.2 Performance on Ambiguous COCO

Table 3 shows the translation performance. It can be found that our models yield a certain amount of gains (in BLEU scores) for En→De translation, and raise both BLEU and METEOR scores for En→Fr. The METER scores for En→De are comparable to that the baseline achieved. However, the improvement is less significant compared to that obtained on Multi30k-16&17 (see Table 1). Considering that the ambiguous COCO contains a larger number of ambiguous words than Multi30k-16&17, we suggest that our method fails to largely shield the baseline from the misleading of ambiguous words.

Nevertheless, our method doesn't result in a two-fold error propagation, but on the contrary it alleviates the negative influences of the errors because:

- Error propagation, in general, is inevitable when a GRU or LSTM unit is used. Both are trained to predict a sequence of words one by one. Appropriate prediction of previous words is crucial for ensuring the correctness of subsequent words. Thus, once a mistake is made at a certain decoding step, the error will be propagated forward, and mislead the prediction of subsequent words.

- The baseline is equipped with a GRU decoder and therefore suffers from error propagation. More seriously, ambiguous words increase the risk of error propagation. This causes a significant performance reduction on Ambiguous COCO. For example, the BLEU score for En→De is 28.7% at best. It is far below that (40.7%) obtained on Multi30k-16&17.

- Two-fold error propagation is suspected to occur when LSTM-based CAP is integrated with the baseline. Though the opposite is actually true. After CAP is used, the translation performance is improved instead of falling down.

6.4 Comparison to the state of the art

We survey the state-of-the-art research activities in the field of MNMT, and compare them with ours

Ambiguous coco (2017)	En→De ($\mu \pm \sigma$/ensemble)			En→Fr ($\mu \pm \sigma$/ensemble)		
	BLEU	METEOR	TER	BLEU	METEOR	TER
Helcl et al (2017)	25.7	45.6	N/A	43.0	62.5	N/A
Caglayan et al (2017)	29.4$^\diamond$	49.2$^\diamond$	N/A	46.2$^\diamond$	66.0$^\diamond$	N/A
Zhou et al (2018)	28.3	48.0	N/A	45.0	64.7	N/A
Baseline	26.4±0.2/28.7	46.8±0.7/48.9	N/A	41.2±1.2/43.3	61.3±0.9/63.3	N/A
+CAP-ATT	27.1±1.2/29.3	47.7±0.9/48.8	**53.0±1.1/50.7**	43.8±1.2/46.8	62.2±0.9/65.0	36.5±1.0/34.5
+CAP-ENC	27.1±1.1/29.4	47.5±0.9/48.7	54.1±1.1/51.2	42.8±1.2/46.3	60.8±0.9/65.3	**38.1±1.0/33.4**
+CAP-DEC	**27.8±1.1/29.9**	**47.8±1.0/49.3**	53.8±1.1/50.8	43.2±1.2/46.1	61.5±0.9/65.3	37.3±1.0/34.3
+CAP-TKN	27.3±1.2/29.6	46.4±0.9/48.9	54.2±1.2/51.1	44.5±1.2/46.8	62.4±0.9/65.3	37.8±1.0/34.0
+CAP-ALL	27.6±1.1/29.8	46.4±0.9/48.9	54.4±1.2/50.8	**44.3±1.2/47.1**	**62.6±0.9/65.4**	36.4±1.0/33.5

Table 3: Performance on Amb-COCO (Note: "$\diamond$" is the sign of the performance when ensemble learning is used.)

En→De	Multi30k-*16*		Multi30k-*17*	
	BLEU	METEOR	BLEU	METEOR
Huang et al (2016)	36.5	54.1		
Calixto et al (2017a)	36.5	55.0		
Calixto et al (2017b)	41.3$^\diamond$	59.2$^\diamond$		
Elliott et al (2017)	40.2$^\diamond$	59.3$^\diamond$		
Helcl et al (2017)	34.6	51.7	28.5	49.2
Caglayan et al (2017a)	41.2$^\diamond$	59.4$^\diamond$	33.5$^\diamond$	53.8$^\diamond$
Helcl et al (2018)	38.7	57.2		
Zhou et al (2018)			31.6	52.2
Ours (μ)	39.6	57.5	33.0	52.2
Ours (ensemble)	**42.1**	**59.9**	**34.3**	**54.5**
En→Fr	Multi30k-*16*		Multi30k-*17*	
	BLEU	METEOR	BLEU	METEOR
Helcl et al (2017)			50.3	67.0
Caglayan et al (2017a)	56.7$^\diamond$	73.0$^\diamond$	55.7$^\diamond$	71.9$^\diamond$
Helcl et al (2018)	60.8	75.1		
Zhou et al (2018)			53.8	70.3
Ours (μ)	60.1	74.3	52.8	68.6
Ours (ensemble)	**63.3**	**77.1**	**56.1**	**71.1**
En→Cz	Multi30k-*16*		Multi30k-*17*	
	BLEU	METEOR	BLEU	METEOR
Helcl et al (2018)	31.0	29.9		
Ours (μ)	32.0	30.2		
Ours (ensemble)	**33.9**	**32.6**		

Table 4: Comparison results on Multi30k (Note: "$\diamond$" is the sign indicating the use of ensemble learning).

(as shown in Table 4). Comparison are made for all the WMT translation scenarios (En→De, Fr and Cz) on Multi30k-16&17 but merely for En→De and En→Fr on ambiguous COCO (as shown in Table 3). To our best knowledge, there is no previous attempt to evaluate the performance of an En→Cz translation model on ambiguous COCO, and thus a precise comparison for that is not available. It is noteworthy that some of the cited work reports the ensemble learning results for MNMT, others make no mention of it. We label the former with a symbol of "$\diamond$" in Tables 3 and 4 to ease the comparison.

It can be observed that our best model outperforms the state of the art for most scenarios over different corpora except the En→Fr case on Multi30k-17. The performance increases are most apparent in the case of En→Fr on Multi30k-16 when ensemble learning is used, where the BLEU and METEOR scores reach the levels of more than 63% and 77%, with the improvements of 6.6% and 4.1%.

We regard the work of Caglayan et al. (2017a) and Calixto et al. (2017a) as the representatives in our systematic analysis. Caglayan et al. (2017a) directly use raw visual features (i.e., V mentioned in section 3.1) to enhance NMT at different stages, including that of initialization, encoding and decoding. Calixto et al. (2017a) develop a doubly-attentive decoder, where both visual features of images and linguistic features of captions are used for computing the attention scores during decoding.

- **Caglayan et al. (2017a)'s model**: Caglayan et al. (2017a)'s model integrates visual features V into the decoding process. By contrast, we conduct the integration using linguistic features which are transformed from visual features. It is proven that our integration approach leads to considerable performance increases. Accordingly, we suppose that reducing incongruence between visual and linguistic features contributes to cross-modality learning in MNMT.

- **Calixto et al. (2017a)'s model**: Our CAP-ATT is similar to Calixto et al. (2017a)'s model due to the use of attention mechanisms during decoding. The difference is that CAP-ATT transforms visual features into linguistic features before attention computation. This operation leads to the increases of both BLEU (2.7%) and METEOR (2%) on Multi30k-16. The results demonstrate that attention scores can be computed more effectively between features of the same type.

6.5 Performance in Adversarial Evaluation

We examine the use efficiency of images for MNMT using Elliott's adversarial evaluation (Elliott, 2018). Elliott suppose that if a model efficiently uses images during MNMT, its performance would degrade when it is cheated by some incongruent images. Table 5 shows the test results, where "C" is specified as a METEOR score which is evaluated when there is not any incongruent image in the test set, while "I" is that when some incogruent images are used to replace the original images.

If the value of "C" is larger than "I", a positive $\triangle_E$-Awareness can be obtained. It illustrates an acceptable use efficiency. On the contrary, a negative $\triangle_E$-Awareness is a warning of low efficiency.

Table 5 shows the test results. It can be observed that our +CAP-ATT and +CAP-ALL models achieve positive $\triangle_E$-Awareness for all the translation scenarios on Multi30k-16. In addition, the models obtain higher values of $\triangle_E$-Awareness than Caglayan et al. (2017a)'s models of `decinit` and `hierattn`. As mentioned above, Caglayan et al directly use visual features to enhance the MNMT, while we use the image-dependent linguistic features that are transformed from visual features. Therefore, we suppose that modality transformation leads to a higher use efficiency of images.

En→De	Multi30k (2016) ($\mu \pm \sigma$)		
	C	I	$\triangle_E$-Awareness
+CAP-ATT	58.5	58.5±0.2	0.001 ±0.002
+CAP-ENC	57.8	58.5±0.1	-0.007 ±0.001
+CAP-DEC	58.3	58.0±0.0	0.020 ±0.001
+CAP-TKN	58.7	58.6±0.1	0.001 ±0.001
+CAP-ALL	**59.0**	**58.5±0.2**	**0.005 ±0.002**
Caglayan et al's trgmul	N/A	N/A	-0.001 ±0.002
Caglayan et al's decinit	N/A	N/A	0.003 ±0.001
Helcl et al's hierattn	N/A	N/A	0.019 ±0.003
En→Fr	Multi30k (2016) ($\mu \pm \sigma$)		
	C	I	$\triangle_E$-Awareness
+CAP-ATT	**74.8**	**74.2±0.1**	**0.005 ±0.001**
+CAP-ENC	73.8	74.2±0.1	-0.004 ±0.001
+CAP-DEC	74.3	74.3±0.1	-0.001 ±0.001
+CAP-TKN	74.9	74.6±0.1	0.003 ±0.001
+CAP-ALL	74.8	74.5±0.1	0.003 ±0.001
En→Cz	Multi30k (2016) ($\mu \pm \sigma$)		
	C	I	$\triangle_E$-Awareness
+CAP-ATT	35.2	34.7±0.2	0.005 ±0.002
+CAP-ENC	34.7	34.4±0.1	0.003 ±0.001
+CAP-DEC	34.8	34.4±0.1	0.004 ±0.001
+CAP-TKN	34.9	35.1±0.1	-0.002 ±0.001
+CAP-ALL	**34.6**	**33.8±0.1**	**0.007 ±0.001**

Table 5: Test results in Elliott's utility test.

7 RELATED WORK

We have mentioned the previous work of MNMT in section 1, where the research interest has been classified into image encoding, encoder-decoder NMT construction and cross-modality learning. Besides, we present the methods of Caglayan et al. (2017a) and Calixto et al. (2017a) in the section 4.4.2, along with the systematic analysis. Besides, many scholars within the research community have made great efforts upon the development of sophisticated NMT architectures, including multi-source (Zoph and Knight, 2016), multi-task (Dong et al., 2015) and multi-way (Firat et al., 2016) NMT, as well as those equipped with attention mechanisms (Sennrich et al., 2015). The research activities are particularly crucial since they broaden the range of cross-modality learning strategies.

Current research interest has concentrated on the incorporation of visual features into NMT (Lala et al., 2018), by means of visual-linguistic context vector concatenation (Libovický et al., 2016), doubly-attentive decoding (Calixto et al., 2017a), hierarchical attention combination (Libovický and Helcl, 2017), cross-attention network (Helcl et al., 2018), gated attention network (Zhang et al., 2019), joint (Zhou et al., 2018) and ensemble (Zheng et al., 2018) learning . In addition, image attention optimization (Delbrouck and Dupont, 2017) and monolingual data expansion (Hitschler et al., 2016) have been proven effective in this field. Ive et al. (2019) use an off-shelf object detector and an additional image dataset (Kuznetsova et al., 2018) to form a bag of category-level object embeddings. Conditioned on the embeddings, Ive et al. (2019) develop a sophisticated MNMT model which integrates self-attention and cross-attention mechanisms into the encoder-decoder based deliberation architecture.

This paper also touches on the research area of image captioning. Mao et al. (2014) provide an interpretable image modeling method using multimodal RNN. Vinyals et al. (2015) design a caption generator (IDG) by Seq2Seq framework. Further, Xu et al. (2015) propose an attention-based IDG.

8 CONCLUSION

We demonstrate that the captioning based harmonization model reduces incongruence between multimodal features. This contributes to the performance improvement of MNMT. It is proven that our method increases the use efficiency of images.

The interesting phenomenon we observed in the experiments is that modality incongruence reduction is more effective in the scenario of En→Fr translation than that of En→De and En→Cz. This raises a problem of adaptation to languges. In the future, we will study on the distinct grammatical and syntactic principles of target languages, as well as their influences on the adaptation. For example, the syntax of French can be considered as most strict. Thus, a sequence-dependent feature vector may be more adaptive to MNMT towards French. Accordingly, we will attempt to develop a generative adversarial network based adaptation enhancement model. The goal is to refine the generated linguistic features by learning to detect and eliminate the features of less adaptability.

Acknowledgements

We thank the reviewers for their insightful comments. The idea is proposed by the corresponding author (Yu Hong). Jian Tang provides an effective assistance for conducting the experiments. We thank the colleagues for their help.

This work was supported by the national Natural Science Foundation of China (NSFC) via Grant Nos. 62076174, 61773276, 61836007.

References

P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, volume 3, page 6.

J. L. Ba, J. R. Kiros, and G. E. Hinton. 2016. Layer normalization. *arXiv:1607.06450*.

D. Bahdanau, K. Cho, and Y. Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv:149.0473*.

O. Caglayan, W. Aransa, A. Bardet, M. García-Martínez, F. Bougares, L. Barrault, M. Masana, L. Herranz, and J. Van de Weijer. 2017a. Lium-cvc submissions for wmt17 multimodal translation task. pages 450–457.

O. Caglayan, W. Aransa, Y. Wang, M. Masana, M. García-Martínez, F. Bougares, L. Barrault, and J. Van de Weijer. 2016. Does multimodality help human and machine for translation and image captioning? *arXiv:1605.09186*.

O. Caglayan, M. García-Martínez, A. Bardet, W. Aransa, F. Bougares, and L. Barrault. 2017b. Nmtpy: A flexible toolkit for advanced neural machine translation systems. *The Prague Bulletin of Mathematical Linguistics*, 109(1):15–28.

I. Calixto, Q. Liu, and N. Campbell. 2017a. Doubly-attentive decoder for multi-modal neural machine translation. *arXiv:1702.01287*.

I. Calixto, Q. Liu, and N. Campbell. 2017b. Incorporating global visual features into attention-based neural machine translation. *arXiv:1701.06521*.

Q. Cao and D. Xiong. 2018. Encoding gated translation memory into neural machine translation. In *EMNLP*, pages 3042–3047.

J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv:1412.3555*.

J. H. Clark, C. Dyer, A. Lavie, Alon, and N. A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *ACL*, pages 176–181. ACL.

J. Delbrouck and S. Dupont. 2017. An empirical study on the effectiveness of images in multimodal neural machine translation. *arXiv:1707.00995*.

M. Denkowski and A. Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *ICML*, pages 376–380.

D. Dong, H. Wu, W. He, D. Yu, and H. Wang. 2015. Multi-task learning for multiple language translation. In *ACL*, pages 1723–1732.

D. Elliott. 2018. Adversarial evaluation of multimodal machine translation. In *EMNLP*, pages 2974–2978.

D. Elliott, S. Frank, L. Barrault, F. Bougares, and L. Specia. 2017. Findings of the second shared task on multimodal machine translation and multilingual image description. *arXiv:1710.07177*.

D. Elliott, S. Frank, K. Sima'an, and L. Specia. 2016. Multi30k: Multilingual english-german image descriptions. *arXiv:1605.00459*.

O. Firat and K. Cho. 2016. Conditional gated recurrent unit with attention mechanism. *System BLEU baseline*, 31.

O. Firat, K. Cho, and Y. Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. *arXiv:1601.01073*.

E. Garmash and C. Monz. 2016. Ensemble learning for multi-source neural machine translation. In *COLING*, pages 1409–1418.

K. He, X. Zhang, S. Ren, and J. Sun. 2016a. Deep residual learning for image recognition. In *CVPR*, pages 770–778.

K. He, X. Zhang, S. Ren, and J. Sun. 2016b. Identity mappings in deep residual networks. In *ECCV*, pages 630–645. Springer.

J. Helcl, J. Libovickỳ, and D. Varis. 2018. Cuni system for the wmt18 multimodal translation task. In *WMT*, pages 616–623.

J. Hitschler, S. Schamoni, and S. Riezler. 2016. Multimodal pivots for image caption translation. *arXiv:1601.03916*.

P. Huang, F. Liu, S. Shiang, J. Oh, and C. Dyer. 2016. Attention-based multimodal neural machine translation. In *WMT*, pages 639–645.

Julia Ive, Pranava Madhyastha, and Lucia Specia. 2019. Distilling translations with visual awareness. *arXiv preprint arXiv:1906.07701*.

N. Kalchbrenner and P. Blunsom. 2013. Recurrent continuous translation models. In *EMNLP*, pages 1700–1709.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, and et al. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL*, pages 177–180. Association for Computational Linguistics.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. 2018. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*.

C. Lala, P. S. Madhyastha, C. Scarton, and L. Specia. 2018. Sheffield submissions for wmt18 multimodal translation shared task. In *ICML*, pages 624–631.

J. Libovickỳ and J. Helcl. 2017. Attention strategies for multi-source sequence-to-sequence learning. *arXiv:1704.06567*.

J. Libovickỳ and J. Helcl. 2018. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. *arXiv:1811.04719*.

J. Libovickỳ, J. Helcl, M. Tlustỳ, P. Pecina, and O. Bojar. 2016. Cuni system for wmt16 automatic post-editing and multimodal translation tasks. *arXiv:1606.07481*.

T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. 2014. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer.

J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. 2014. Explain images with multimodal recurrent neural networks. *arXiv:1410.1090*.

J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. 2011. Multimodal deep learning. In *ICML*, pages 689–696.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318. Association for Computational Linguistics.

S. Ren, K. He, R. Girshick, and J. Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, and et al. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.

R. Sennrich, B. Haddow, and A. Birch. 2015. Neural machine translation of rare words with subword units. *arXiv:1508.07909*.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *ICML*, volume 200.

L. Specia, S. Frank, K. Sima'an, and D. Elliott. 2016. A shared task on multimodal machine translation and crosslingual image description. In *ICML*, volume 2, pages 543–553.

I. Sutskever, O. Vinyals, and Q. V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112.

O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. 2015. Show and tell: A neural image caption generator. In *CVPR*, pages 3156–3164.

K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pages 2048–2057.

P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *ACL*, 2:67–78.

Zhuosheng Zhang, Kehai Chen, Rui Wang, Masao Utiyama, Eiichiro Sumita, Zuchao Li, and Hai Zhao. 2019. Neural machine translation with universal visual representation. In *International Conference on Learning Representations*.

R. Zheng, Y. Yang, M. Ma, and L. Huang. 2018. Ensemble sequence level training for multimodal mt: Osu-baidu wmt18 multimodal machine translation system report. *arXiv:1808.10592*.

M. Zhou, R. Cheng, Y. J. Lee, and Z. Yu. 2018. A visual attention grounding neural model for multimodal machine translation. *arXiv:1808.08266*.

B. Zoph and K. Knight. 2016. Multi-source neural translation. *arXiv:1601.00710*.

Error Causal inference for Multi-Fusion models

Chengxi Li
University of Kentucky
Lexington, KY 40506
chengxili@uky.edu

Brent Harrison
University of Kentucky
Lexington, KY 40506
harrison@cs.uky.edu

Abstract

In this paper, we propose an error causal inference method that could be used for finding dominant features for a faulty instance under a well-trained multi-modality input model, which could apply to any testing instance. We evaluate our method using a well-trained multi-modalities stylish caption generation model and find those causal inferences that could provide us the insights for next step optimization.

1 Introduction

As machine learning models become more complex and training data become bigger, it is harder for humans to find errors manually once some output went wrong. This problem is exacerbated by the black box nature of most machine learning models. When a model fails, it can be difficult to determine where the error comes from. This is especially true in problems that are inherently multimodal, such as image captioning, where often multiple models are combined together in order to produce an output. This lack of transparency or ability to perform a vulnerability analysis can be a major hindrance to machine learning practitioners when faced with a model that doesn't perform as expected.

Recently, more and more people begin to fuse text and visual information for downstreaming task. In many cases, these models utilize specialized, pre-trained models to extract features. In these situations, it is highly likely that the source of these errors is from these pre-trained networks either being misused or not being interpreted correctly by the larger machine learning architecture. In this paper, we explore how one would perform a vulnerability analysis in these situations. Specifically, we are interested in identifying model errors likely caused by these pre-trained networks. Specifically, we aim to diagnose these errors by systematically removing elements of the larger machine learning model to pinpoint what the causes of errors happen to be. This is especially critical in tasks that utilize multi-modality input models since often these models utilize attention. If the model attends to the wrong features, then this error could potentially cascade throughout the network. In other words, we seek to answer the question, "Given a trained model M which has input features x, y, z, if the current test example is not performing well, is that because of the given features or not? If it is, which specific feature is more likely to blame?"

By answering this question, we can give machine learning practitioners, specifically those who are inexperienced with machine learning and AI concepts, some direction in how to improve the performance of their architecture. We summarize our contributions as follows: 1. we provide a practical method to discover causal errors for multi-modality input ML models; 2. we explore how this method can be applied to state-of-the-art machine learning models for performing stylish image captioning; 3. Evaluate our method by through a case study in which we assess whether we can improve the performance of the investigated instance by removing or replacing these problematic features.

2 Related Work

Our approach to sourcing these errors uses causal inference (Peters et al., 2016; Hernán and Robins, 2020). In this section, we will review works related to causal inference as well as works that provided the inspiration for this paper.

Invariance Principle Invariance principle has been used for finding general causal for some outcome under designed treatment process, where people desired to find actual effect of a specific phenomenon. Invariant causal prediction (Peters et al., 2016) has been proposed to offer an practical way to find casuals under linear model assumption. It later got extended to nonlinear model and data (Heinze-Deml et al., 2018). This invariance can be roughly phrased as the outcome Y of some model

Proceedings of the Second Workshop on Advances in Language and Vision Research, pages 11–15

M would not change due to environment change once given the cause for this Y. An example of an *environment change* when $Y = M(X, Z, W)$ and the cause for Y is X, could be a change on Z or W. The invariance principle has been popularly used in machine learning models to train causal models (Arjovsky et al., 2019; Rojas-Carulla et al., 2018). We are going to employ the same insight, using the invariance principle to find cause in our paper but landing in different perspectives. We are not intended to train a model, instead, we are going to use the well-trained models to derive the source cause for lower performance instances.

Potential Outcome and Counterfactual. (Rubin, 2005) proposed using potential outcomes for estimating causal effects. Potential outcomes present the values of the outcome variable Y for each case at a particular point in time after certain actions. But usually, we can only observe one of the potential outcome since situations are based on executing mutually exclusive actions (e.g. give the treatment or don't give the treatment). The unobserved outcome is called the "counterfactual" outcome. In this paper we can observe the counterfactual by removing certain input features from the language generation based on multi-input task.

Debugging Errors Caused by Feature Deficiencies This paper is also related to debugging errors from input. While we are more focus on using a causal inference way to get the real cause for low performance rather than only exploring associations (Amershi et al., 2015; Kang et al., 2018)

3 Methodology

The goal of this paper is to perform a causal analysis in order to determine the likely source of errors in a well-trained model. In the following sections, we will outline our underlying hypotheses related to this task and go into details on the task itself.

3.1 Hypothesis

Hypothesis 1: *With a fixed model, if the output of an instance k is unchanged after an intervention, I, then this is called **output invariance**. The causes of the output for this instance k are irrelevant to the features associated the intervention, I.*
Using this output invariance principle, we can identify features that are irrelevant to the prediction made. After removing these irrelevant features, the ones that remain should contribute to any errors present in the output. Given the strictness of the output invariance principle, it is often the case that very few features are identified as the cause of any error present. In some cases, no features are identified. In this paper, we are interested in determining the cause of errors by masking out certain features, specifically those that are unlikely to be the cause of an error. As such, we are interested in the specific case where the removal of certain features does not cause the performance of the model to improve. This phenomenon, which we refer to as ***output non-increasing*** will be rephrased below.
Hypothesis 2: *With a fixed model, if the output of an instance, k, after an intervention, I, is either less than or equal to the original performance of instance k, then this is called **output non-increasing**. Then, the features associated with intervention, I, are likely irrelevant to the cause of any error.*
In this paper, we specifically perform interventions that involving masking/hammering out certain input features. *Hammering out* features could mean zero out input features or specific weights, or even remove certain input modalities, etc.. In this paper we will change the values of certain input features f to 0. Then, output is regenerated according to this new input. If the output is unchanged (or gets worse), then we will remove this feature f from the causal features list. Before we perform these interventions, we first want to identify the errors which do not relate to any of these features. This leads to the next hypothesis.
Hypothesis 3: *If we hammered out all input features and output invariance still holds for instance k, we will record the cause for instance k having lower performance as being due to model and dataset bias. We will refer to this as **bias error**.*
In this paper, we are interested in more than bias errors. With this goal, we arrive at our final hypothesis on performing causal inference for identifying errors.
Hypothesis 4: *If the performance of instance k is poor and the output of instance k is not caused by bias errors, and if all interventions keep feature $f^\star$ unchanged and we still have output non-increasing, we will say $f^\star$ is the error feature which causes the lower performance output for k.*
With all of the above hypotheses we can infer whether the low performance of the instance k is caused by a single feature f or not. Next we will show how these hypotheses can be utilized to identify the causes of errors.

3.2 Causal Graph: with and without Hammering out Features

As we know, when we build a model in deep learning, we always assume a casual graph in advance and then fit data into the graph for training. Figure 1 shows a sample causal graph (a) with multiple input features. These features will be fit into a black box model and finally the model will, in this case, generate some set of output text. Once we have finished training, we will be able to deploy the model and see each testing instance's performance. With a well-trained model, we can perform many interventions, or investigate specific features by intervening on them in different environments. In practice, however, it is impossible for us to obtain all the random environments. Based hypotheses 3 and 4, along with the steps that people take to perform causal predictions in linear (Peters et al., 2016) and non-linear (Heinze-Deml et al., 2018) situations, we, in this paper, give a more detailed and practical definition below to help us identify whether a feature set S is the causal feature set or not when an instance k having error and this error is not a bias error defined in hypothesis 3. Here S could be a feature set composed of a single feature or multiple features. After hamming out some features, we call a remaining feature set P as S's **parental set** when $S \subset P$. We denote $\mathcal{F}_S$ as: $\mathcal{F}_S = \{g(P) \mid P \text{ is a parental set of } S\}$ and

$$g(P) = \begin{cases} P & \text{if } P \text{ satisfies output} \\ & \text{non-increasing,} \\ \emptyset & \text{otherwise.} \end{cases}$$

Then we could extract the estimated causal feature set $\hat{F}$ as:

$$\hat{F} = \bigcap_{F:F\in\mathcal{F}_S} F \tag{1}$$

To simply understand above, we basically check all the parental sets of S on output non-increasing property to finally make decision on whether S is an error casual feature set or not. In this paper, we mainly focus on evaluating single feature set. To better explain, we also display all the interventions (b-h in Figure 1) we have done to the features (masking out some features) when there are a total of 3 features in the assumed causal graph. We will infer the causal feature for a low performance instance k based on all of these potential outcomes before and after interventions. We will use $o_x, x \in \{a, b, c...h\}$ to note the score for output

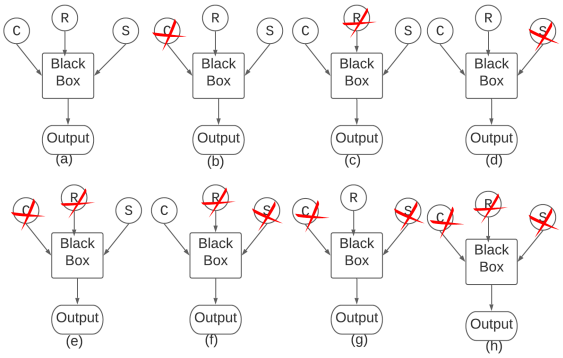


Figure 1: Displays the causal graph with various sets of features zeroed out and a red cross mark signifies zeroing out

generation of graphs in Figure 1. First, we extract the instances when the error cause is independent of any features where we find all the instances $\mathcal{B}$, which satisfy $o_a == o_h$. Then the following causal feature inferences will exclude detected instances in $\mathcal{B}$ first. As we are specifically interested in single feature errors, we will enumerate the situation when causal features are R, C, S for instance k, respectively. First of all, according to hypothesis 3 and 4, k $\notin \mathcal{B}$. The causal feature is S when: $o_a >= o_b >= o_c >= o_e$; The causal feature is C when: $o_a >= o_c >= o_d >= o_f$; The causal feature is R when: $o_a >= o_b >= o_d >= o_g$.

It is important to note that removing an error feature does not necessarily mean that the performance will increase, as it is possible that there are other sources of errors that still keep performance low. In these rules, we use the "=" sign in its strictest sense. However, one can always use it in a way that is interchangeable with "very similar." For example, if the difference between two output scores is 10^{-16}, you can choose to regard these two scores as equal per your needs.

4 Experiment

To show the effectiveness of our approach, we will examine its performance on a stylish image captioning task that uses multi-modality feature fusion. While we focus on this task as an example, this approach could be applied to many other domains.

4.1 Dataset

We have chosen the dataset and the task based on three qualities:the work has a well-trained saved model which we could use for intervention and inference; this work still has room to be improved

by identifying and removing the source of potential errors; the work utilizes multiple input features in a way that enables removing said features.

Specifically, we choose the work on the 3M model (Li and Harrison, 2021) for stylish text captioning. We do this because it relies on generating captions using several input features including pre-trained text features (C), ResNext features (R), and style information (S) as an input. We would like to explore whether these input features have caused problems when instances are under performing. The dataset we examine is the test set from the PERSONALITY-CAPTIONS dataset (Shuster et al., 2019) using in Li and Harrison's work along with the pretrained model they provide [1]. Even though we use its test set in our experiment, our method could be applied on any set of data of any size when there is a debugging need for multi-fusion models. We will leave this for future work.

4.2 Implementing details

Specifically, we define an instance is under performance in 3M (Li and Harrison, 2021) when the BLEU1 (Papineni et al., 2002) score is lower than the median BLEU1 value among all testing data. In total, we have investigated 9981 instances and 4982 of them are classified as under performing. 74 of these have been detected as bias errors. So finally, 4908 instances have been examined for single feature errors.

We first perform causal inference for style feature and denote those instances that have style error as K_s. Then perform the causal finding steps for ResNext and dense captions without differentiating the order in the remaining instances. The reason to decide such order is due to the structure of 3M, where style is used globally to refine ResNext and dense captions for later stylish text generations while ResNext and dense caption have the same importance for text generations.

4.3 Evaluation

The reason to find the cause for the errors is that we would like to further improve a model when it is well-trained or make a remedy when the model is malfunctioned, especially from the source side. Thus, we evaluate casual predictions by evaluating whether we could improve the model's performance by just altering the causal feature. There are many potential treatments that we could make on the source side such as data augmentation, feature replacement, or feature removal. For each instance k with predicted causal feature f, if its performance could be improved by improving f, then we will judge the causal error inference for this instance k as correct, otherwise incorrect. More details on the specific interventions we use are outlined below: Style: (S1) replace current style with 5 other well-trained styles S, where most instances with style s, $s \in S$ has better BLEU1 score than the medium BLEU1 score. (S2) remove Style. Dense Caption: (C1) replace dense caption to ground truth; (C2) remove dense caption. Resnext: (R1) replace dense caption to ground truth and then remove Resnext, where we make sure at least one of the visual features is valid. (R2) remove Resnext.

We will record the best output BLEU1 score after each intervention. If the intervention results in a higher BLEU1 score than the output prior to the intervention, then the feature in question will be marked as the cause of an error. For all the instances which have been ascribed by a feature f, we calculate the percentage of those in which the BLEU1 score could be improved and report them in Table 1.

5 Result and Discussion

The result is shown in the Table 1. We see that for each feature, most of the instances have increased their performance by improving the predicted features. This performance is also a conservative value as we only did limited feature improvements. For example, for Resnext, we have no better features available and, thus, could not do a replacement. Also in Table 1, the style feature is the most predicted error causal feature among all three feature modalities. We have 1041 instances point its performance error towards style. We speculate this is resulting from the weak training of a certain set of styles, since the BLEU score can be improved if replaced with other better-trained styles for 89.4% of these instances. To further investigate this, we report the frequency of the styles in those 1041 instances in Figure 2 and intend to see whether the estimated error styles are distributed sparsely (all styles are not trained well) or densely (a certain set of styles is not trained well). From Figure 2 we can see that many styles repeatedly appear as errors for various instances, which aligns our speculation. With these predicted error styles, we can either do

[1] https://github.com/cici-ai-club/3M

Table 1: The evaluation result for each feature under casual inference. Predictions Count are the number of instances predicted with corresponding feature errors.

Feature	Predictions Count	Improvement(%)
Style	1041	0.894
Dense Caption	378	0.797
Resnext	300	0.769

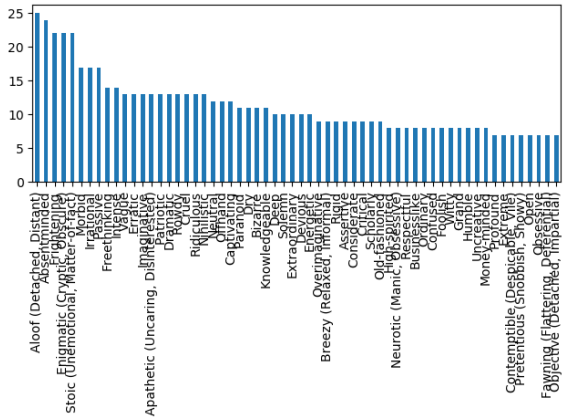


Figure 2: The styles are those frequently predicted as the causal errors; the horizontal bar represents the frequency. Here we select the top 50 styles.

some data augmentation to cover the gap between training and testing or redesign the training process to enable the model to focus more intently on these styles.

6 Conclusion

In this paper, we apply an extended invariance principle to provide a method for performing error causal inference. We evaluate our method under on a stylish image captioning model that uses multi-modal fusion in its input features. We show that we could improve the performance of this model based on simply removing or replacing those found causal errors. Over 70% of the predicted errors could be modified to improve performance. Also, our method is model-agnostic, it could be used for different fusion model for optimization, debugging or assessing purpose.

References

Saleema Amershi, Max Chickering, Steven M Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. 2015. Modeltracker: Redesigning performance analysis tools for machine learning. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 337–346.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Christina Heinze-Deml, Jonas Peters, and Nicolai Meinshausen. 2018. Invariant causal prediction for nonlinear models. *Journal of Causal Inference*, 6(2).

Miguel A Hernán and James M Robins. 2020. Causal inference: what if.

Daniel Kang, Deepti Raghavan, Peter Bailis, and Matei Zaharia. 2018. Model assertions for debugging machine learning. In *NeurIPS MLSys Workshop*.

Chengxi Li and Brent Harrison. 2021. 3m: Multi-style image caption generation using multi-modality features under multi-updown model.

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

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. 2016. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pages 947–1012.

Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. 2018. Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1):1309–1342.

Donald B Rubin. 2005. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331.

Kurt Shuster, Samuel Humeau, Hexiang Hu, Antoine Bordes, and Jason Weston. 2019. Engaging image captioning via personality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12516–12526.

Leveraging Partial Dependency Trees to Control Image Captions

Wenjie Zhong
The University of Tokyo
zvengin@is.s.u-tokyo.ac.jp

Yusuke Miyao
The University of Tokyo
yusuke@is.s.u-tokyo.ac.jp

Abstract

Controlling the generation of image captions attracts lots of attention recently. In this paper, we propose a framework leveraging *partial syntactic dependency trees* as control signals to make image captions include specified words and their syntactic structures. To achieve this purpose, we propose a ***S**yntactic **D**ependency **S**tructure **A**ware **M**odel* (**SDSAM**), which explicitly learns to generate the syntactic structures of image captions to include given partial dependency trees. In addition, we come up with a metric to evaluate how many specified words and their syntactic dependencies are included in generated captions. We carry out experiments on two standard datasets: Microsoft COCO and Flickr30k. Empirical results show that image captions generated by our model are effectively controlled in terms of specified words and their syntactic structures. The code is available on GitHub[1].

1 Introduction

Controllable image captioning emerges as a popular research topic in recent years. Existing works attempt to enhance models' controllability and captions' diversity by controlling the attributes of image captions such as style (Mathews et al., 2016), sentiments (Gan et al., 2017), contents (Dai et al., 2018; Cornia et al., 2019; Zhong et al., 2020) and part-of-speech (Deshpande et al., 2019). However, some important attributes of image captions like words and syntactic structures, are ignored in previous works. For example, for the image in the Figure 2, the work (Cornia et al., 2019) specifies a set of objects like 'dog, man, frisebee' as a control signal, but there still exist lots of possibilities of composing them into different captions, such as 'a dog and a man play frisebee on grass' and 'a dog playing with a man catches frisebee', since both words and syntactic structures are not determined yet.

[1]https://github.com/ZVengin/DepControl_ALVR

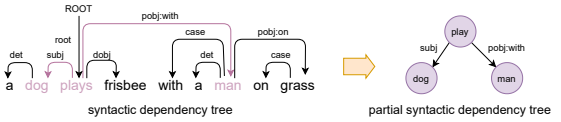


Figure 1: An example of syntactic dependency tree(left) and partial dependency tree (right)

To address this challenging issue, we propose a framework, which employs *partial dependency trees* as control signals. As shown in Figure 1, a partial dependency tree, a sub-tree of a syntactic dependency tree, contains words and their syntactic structures, and thus we can utilize it to specify control information about words and their syntactic structures.

In addition, we develop a pipeline model called *syntactic dependency structure-aware model* (SDSAM) which first derives a full syntactic dependency tree and then flatten it into a caption. The motivation behind this pipeline model is that we assume explicitly generating syntactic dependency trees as intermediate representations can better help the model learn how to apply the specified syntactic information to the captions and the intermediate representations can give users an intuitive impression on which part of the captions' syntactic structures is controlled.

Finally, we propose a syntactic dependency-based evaluation metric which evaluates whether the generated captions have been controlled in terms of syntactic structures. Our metric is computed based on the overlap of syntactic dependencies which is different from existing metrics like BLEU (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2014), ROUGE (Lin, 2004), CIDEr (Vedantam et al., 2015) and SPICE (Anderson et al., 2018) which rely on the overlap of n-grams or semantic graphs. Empirical results show that image captions generated by our model are effectively controlled in terms of specified words and their syntactic structures.

Proceedings of the Second Workshop on Advances in Language and Vision Research, pages 16–21

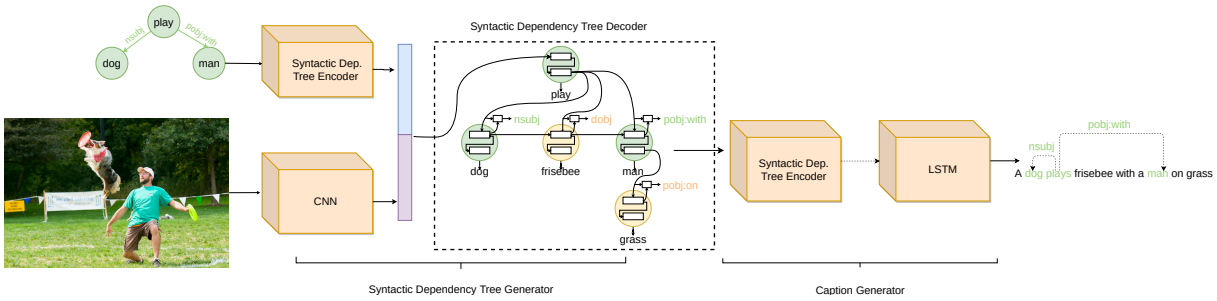


Figure 2: Model architecture: our model generates captions in two steps: (1) generating syntactic dependency tree using syntactic dependency tree generator. (2) flatting it into a caption using caption generator.

2 Framework Definition

The task presented in this paper is defined as generating a caption sentence (i.e. word sequence) $y = \langle w_1, \cdots, w_{|y|} \rangle$ given an image I and a partial dependency tree P as input, so that the dependency tree T_y of y includes P as far as possible. The syntactic dependency tree of a sentence, as shown in Figure 1, refers to a tree structure to represent syntactic relations between words. A syntactic dependency tree T_x of a sentence $x = \langle w_1, \cdots, w_{|x|} \rangle$ is defined as a set of dependencies, $\{D_1, D_2, \cdots, D_{|T_x|}\}$, where $|T_x|$ denotes the number of dependencies in T_x. Each dependency D_k is expressed in the form of $w_i \xrightarrow{e_{i,j}} w_j$, where w_i and w_j are the head word and the dependent word of D_k, and $e_{i,j}$ is the dependency label. We denote child nodes of w_i as $C(w_i)$; i.e. $C(w_i) = \{w_j | w_i \xrightarrow{e_{i,j}} w_j \in T_x\}$. A partial dependency tree P here refers to a sub-tree of the syntactic dependency tree of some sentence.That is, $P \subseteq T_x$ for some sentence x.

3 Syntactic Dependency Structure Aware Model

The syntactic dependency structure-aware model(SDSAM) shown in Figure 2 generates image captions in two steps: (1) the syntactic dependency tree generator on the left part derives a full syntactic dependency tree from the image and the partial dependency tree. (2) the caption generator on the right part will flatten the syntactic dependency tree into a caption.

The Syntactic Dependency Tree Generator The syntactic dependency tree generator encodes the input image with a CNN network implemented with Resent152 (He et al., 2016) into image features and encodes the partial dependency tree with a syntactic dependency tree encoder implemented with Tree-LSTM (Tai et al., 2015) into partial dependency tree features.

After combining the image features and the partial dependency tree features, the syntactic dependency tree generator derives the full syntactic dependency tree using the syntactic dependency tree decoder from the combined features $\boldsymbol{s}$. The syntactic dependency tree decoder consists of two attention modules, Attn_{in} and Attn_{out}, and two interweaved GRU networks (Cho et al., 2014), GRU_v and GRU_h. The decoding process is carried out from the root node to leaf nodes in a top-down manner. For a node w_i, its child nodes are decoded one by one from left-to-right. Each child node is predicted based on the information of its parent node and its left sibling node generated in previous steps. At the mean while, the attention modules highlight the words to be generated for the current child node. Assuming we decode the child w_j of node w_i, the hidden state of node w_i and node w_j are denoted as $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ respectively. The left sibling of node w_j is denoted as w_{j-1} and its hidden state as $\boldsymbol{h}_{j-1}$. For each input image, we detect a set of keywords $c = \{r_1, \cdots, r_{|c|}\}$ following the method proposed in (You et al., 2016), and encode c into a matrix $\boldsymbol{C} \in \mathbb{R}^{E_w \times |c|}$, where E_w is the size of word embedding.

$$\boldsymbol{h}_0 = \boldsymbol{U}^{(s)} \boldsymbol{s} \tag{1}$$

$$\tilde{\boldsymbol{h}}_i = \text{GRU}_v(\boldsymbol{h}_i, \boldsymbol{w}_i) \tag{2}$$

$$\boldsymbol{c}_{\text{in}} = \text{Attn}_{\text{in}}(\boldsymbol{w}_i, \boldsymbol{C}) \tag{3}$$

$$\boldsymbol{h}_j = \text{GRU}_h(\tilde{\boldsymbol{h}}_i, [\boldsymbol{h}_{j-1}; \boldsymbol{w}_{j-1}; \boldsymbol{c}_{\text{in}}]) \tag{4}$$

$$\boldsymbol{c}_{\text{out}} = \text{Attn}_{\text{out}}(\boldsymbol{h}_j, \boldsymbol{C}) \tag{5}$$

$$w_j \sim \text{Softmax}(\boldsymbol{U}^{(w)} \boldsymbol{h}_j + \boldsymbol{V}^{(w)} \boldsymbol{c}_{\text{out}}) \tag{6}$$

$$e_{i,j} \sim \text{Softmax}(\boldsymbol{U}^{(e)} \boldsymbol{h}_j + \boldsymbol{V}^{(e)} \tilde{\boldsymbol{h}}_i) \tag{7}$$

where:

$$\text{Attn}(\boldsymbol{q}, \boldsymbol{C}) = \boldsymbol{C}\boldsymbol{\alpha} \tag{8}$$

$$\boldsymbol{\alpha} = \text{Softmax}(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{v}) \tag{9}$$

$$\boldsymbol{A} = \tanh(\boldsymbol{U}^{(\alpha)}(\boldsymbol{q} \cdot \boldsymbol{1}^{\mathsf{T}}) + \boldsymbol{V}^{(\alpha)}\boldsymbol{C}) \tag{10}$$

In the above formulas, $\boldsymbol{U}^{(s)} \in \mathbb{R}^{H \times E_s}$, $\boldsymbol{U}^{(w)} \in \mathbb{R}^{V_w \times H}$, $\boldsymbol{U}^{(e)} \in \mathbb{R}^{V_e \times H}$,$\boldsymbol{U}^{(\alpha)} \in \mathbb{R}^{E_a \times E_q}$, $\boldsymbol{V}^{(w)} \in \mathbb{R}^{V_w \times H}$, $\boldsymbol{V}^{(e)} \in \mathbb{R}^{V_e \times H}$ and $\boldsymbol{V}^{(\alpha)} \in \mathbb{R}^{E_a \times E_w}$ are parameters for reshaping features. Here E_s, E_a and E_q are the size of the input feature $\boldsymbol{s}$, the attention feature $\boldsymbol{A}$ and the query $\boldsymbol{q}$ respectively. V_w and V_e are the vocabulary size for the node and edge respectively and H is the size of hidden states. In equation (10), $\boldsymbol{v} \in \mathbb{R}^{E_a \times 1}$ is a parameter and $\boldsymbol{1} \in \mathbb{R}^{|c| \times 1}$ is a vector with all elements being one.

The Caption Generator The caption generator takes the syntactic dependency tree generated in the first step as input and encodes it with the syntactic dependency tree encoder into syntactic dependency tree features. The caption generator combines it with image features extracted in the first step and use the combined features to initialize the LSTM decoder (Hochreiter and Schmidhuber, 1997) to generate the caption.

4 Experiment

Preparing Datasets with Partial Dependency Trees For evaluation, we apply two methods to create partial dependency trees for on Microsoft COCO (Chen et al., 2015) and Flickr30k (Young et al., 2014). The first method extracts partial dependency trees from reference captions. We parsing reference captions to syntactic dependency trees using Spacy [2] and then randomly sample subsets from each syntactic dependency tree. Sampled partial dependency trees are then paired with corresponding reference captions. The dataset created by this procedure is denoted as $test_{gold}$ in Section 5.

The other method creates partial dependency trees from images in two steps: (1) we first train a syntactic dependency classifier to predict syntactic dependencies for an input image. (2) Predicted syntactic dependencies are combined to form a syntactic dependency graph for the input image, from which partial dependency trees are sampled. The dataset created by this procedure is denoted as $test_{pred}$ in Section 5.

[2]https://spacy.io

For training, following the first method, we directly sample a partial dependency tree from one of the reference captions for each image and the paired reference caption is used as a training target.

Evaluation Metric The evaluation metrics for image captioning fall into two categories: (1)Quality: evaluating the relevance to human annotations with metrics including BLEU (Papineni et al., 2002) and METEOR (Denkowski and Lavie, 2014); ROUGE (Lin, 2004), and CIDEr (Vedantam et al., 2015) and SPICE (Anderson et al., 2018). (2)Control-ability: evaluating whether generated image captions are successfully controlled by partial dependency trees. We devise a new metric called ***D**ependency **B**ased **E**valuation **M**etric* (**DBEM**) for this purpose. Assuming that a partial dependency tree $P = \{D_1, \cdots, D_{|P|}\}$ is input, DBEM calculates how many syntactic dependencies specified in the partial dependency tree are included in the dependency tree T_y of generated caption y. The DBEM score for the evaluation dataset is given as an average of this score for each input. Formally,

$$DBEM(P, T_y) = \frac{\sum_{D \in P} \mathbf{1}(D, T_y)}{|P|}, \tag{11}$$

$$\mathbf{1}(D, T) = \begin{cases} 1 & \text{if } D \in T \\ 0 & \text{if } D \notin T. \end{cases} \tag{12}$$

Experiment Setting The training of our model is split into two stages including training the syntactic dependency tree generator and training the caption generator. We set the size of hidden states to be 512, the word embedding size to be 512, and the dependency label embedding size to be 300. We train our model using the Adam optimizer (Kingma and Ba, 2015) with a learning rate $5e^{-4}$ for the first stage and $1e^{-4}$ for the second stage. Two models, including our SDSAM model and the NIC model (Vinyals et al., 2015) with its encoder being replaced with Resnet152, are compared under three different control inputs. (1) *None* control: input is an image. (2) *Half* control: input is an image and the words of a partial dependency tree. (3) *Full* control: input is an image and a partial dependency tree.

5 Results and Analysis

Quality (1) Results on $\text{test}_{\text{gold}}$: We show BLEU-4 (B4), METEOR (M), ROUGE (R), CIDEr (C), and SPICE (S) scores on $test_{gold}$ in Table 1, whose

Control	Model	Microsoft COCO					Flickr30k				
		B-4	M	R	C	S	B-4	M	R	C	S
None	NIC	9.3	15.5	35.6	88.5	21.7	5.9	11.4	28.6	36.3	14.1
	SDSAM	9.6	16.0	35.5	94.4	23.7	4.6	10.8	25.8	34.9	14.8
Half	NIC	25.5	27.3	52.2	232.2	41.9	12.7	18.3	38.3	88.7	26.4
	SDSAM	24.7	26.6	52.6	234.4	44.1	12.4	18.4	40.2	103.8	32.8
Full	NIC	32.5	29.9	58.3	294.1	47.1	15.0	19.1	41.0	105.5	27.7
	SDSAM	30.2	29.2	57.1	282.3	48.4	13.4	18.9	41.5	114.2	33.7

Table 1: Evaluation of quality on $test_{gold}$. Each generated caption is only evaluated against its corresponding reference caption.

Control	Model	Microsoft COCO					Flickr30k				
		B-4	M	R	C	S	B-4	M	R	C	S
None	NIC	27.2	23.8	51.2	86.7	17.1	18.1	18.1	42.8	35.3	11.5
	SDSAM	28.0	24.5	50.9	90.2	18.0	15.8	17.5	39.7	35.6	11.7
Half	NIC	27.9	24.4	52.0	88.4	18.3	18.2	17.6	42.3	32.1	11.9
	SDSAM	26.8	24.4	51.1	88.7	18.5	17.2	17.4	40.5	32.9	12.3
Full	NIC	25.6	24.6	51.0	86.5	18.6	15.7	18.4	41.5	31.2	12.5
	SDSAM	26.0	24.5	50.8	87.7	19.0	16.7	17.7	40.7	32.4	12.4

Table 2: Evaluation of quality on $test_{pred}$. Each generated caption is evaluated against all reference captions of its corresponding image.

partial dependency trees are sampled from reference captions. This table shows that both NIC and SDSAM achieve significant improvements on evaluation scores when more control signals are input. This indicates that generated captions become closer to reference captions. These improvements are expectable since control signals contain information of reference captions. This result attests that partial dependency trees carry information useful for generating specific sentences. When both models are given the same control signals, SDSAM has comparable performance to NIC in n-gram based metrics (i.e. BLEU-4, METEOR, ROUGE and CIDEr), while achieving a significantly better performance on SPICE, which is a semantic relation based metric. This result reveals an interesting phenomenon that explicitly learning the syntactic structures of captions can improve performance on the semantic relation based metric.

(2) Results on $\text{test}_{\text{pred}}$: We show the evaluation results on $test_{pred}$ in Table 2, whose partial dependency trees are generated from images. For NIC and SDSAM, evaluation scores mostly remain the same level, but slight improvements are observed in SPICE. This result reveals that partial dependency trees generated from images do not have a significant impact on the quality of image captions, while giving partial dependency trees as control signals do not harm caption quality. For the same control signals, SDSAM has a better performance on SPICE in most cases, which follows the results on $test_{gold}$.

Controllability DBEM scores on $test_{gold}$ and $test_{pred}$ are shown in Table 3. The table shows that the DBEM scores of both models are very low when no control is given. This reveals that only a small proportion of syntactic dependencies in partial dependency trees appear in reference captions by chance, indicating that additional input to control syntactic structures is meaningful. When the models are given words as control signals, the DBEM scores are significantly increased, meaning that both models can infer syntactic structures from words even without explicit syntactic structure information. However, it is also clear that nearly half of the specified dependencies are missing in generated captions. These observations suggest that words provide useful information as control signals, but are insufficient to specify syntactic structures completely. When partial dependency trees are input, the DBEM scores further improve significantly. It means that most syntactic dependencies specified in partial dependency trees are included in generated captions. This result demonstrates that syntactic structure information plays an important role in precisely controlling image captions.

When the models are given no control signals, SDSAM has better DBEM scores than NIC. This is possibly because SDSAM explicitly learns to generate syntactic dependency trees, and can bet-

Control	Model	$test_{gold}$		$test_{pred}$	
		MSCOCO	Flickr30k	MSCOCO	Flickr30k
None	NIC	7.1	4.5	12.2	15.5
	SDSAM	9.5	5.4	19.6	19.8
Half	NIC	47.8	33.9	61.4	64.7
	SDSAM	51.4	44.6	64.2	72.3
Full	NIC	68.3	42.7	86.2	85.0
	SDSAM	69.5	52.9	87.1	87.5

Table 3: Evaluation of controllability (DBEM scores)

ter generate high-frequency syntactic dependencies that also frequently appear in partial dependency trees. When the models are given words and/or syntactic dependencies as control signals, SDSAM achieves higher DBEM scores than NIC. This result demonstrates that explicitly learning to generate syntactic dependency trees as an intermediate representation contributes to better controlling of image captions.

6 Case Study

In Figure 3, we show an example of the output from our model on $test_{pred}$. Our syntactic dependency classifier first predicts a syntactic dependency graph from the input image. Once the syntactic dependency graph is constructed, we sample three partial dependency trees with different node numbers as shown in the figure. Finally, our SDSAM model infers the captions from the input image and the partial dependency trees. From this example, it is obvious that all words and syntactic structures specified in partial dependency trees also appear in the generated captions. Furthermore, the three generated captions are considerably different from each other, demonstrating that giving partial dependency trees as control signals can improve captions' diversity.

7 Conclusion

We presented a framework for controlling image captions in terms of words and syntactic structures by giving partial dependency trees as control signals. We develop a syntactic dependency structure aware model to explicitly learn the syntactic structures in control signals. Empirical results show that image captions generated by our model are effectively controlled in terms of specified words and their syntactic structures. Furthermore, the results indicate that explicitly learning to generate the syntactic dependency trees of captions enhances the model's controllability.

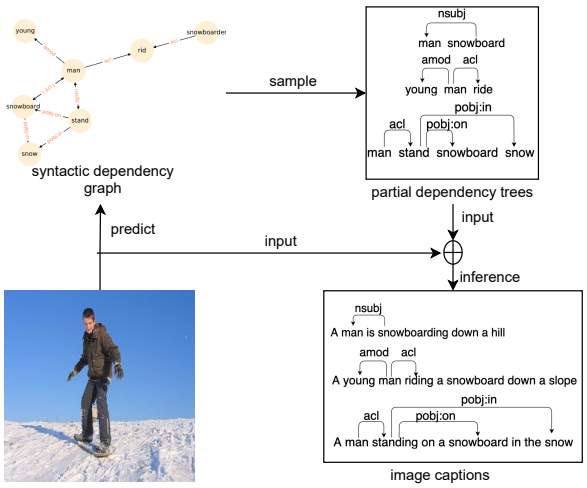


Figure 3: Case study: This figure shows an example generated during inference phase.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6077–6086. IEEE Computer Society.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. 2015. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar*, pages 1724–1734. Association for Computational Linguistics.

Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. 2019. Show, control and tell: A framework for generating controllable and grounded captions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8307–8316. IEEE Computer Society.

Bo Dai, Sanja Fidler, and Dahua Lin. 2018. A neural compositional paradigm for image captioning. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 656–666.

Michael J. Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation eval-

uation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA*, pages 376–380. The Association for Computer Linguistics.

Aditya Deshpande, Jyoti Aneja, Liwei Wang, Alexander G. Schwing, and David A. Forsyth. 2019. Fast, diverse and accurate image captioning guided by part-of-speech. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 10695–10704. IEEE Computer Society.

Chuang Gan, Zhe Gan, Xiaodong He, Jianfeng Gao, and Li Deng. 2017. Stylenet: Generating attractive visual captions with styles. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 955–964. IEEE Computer Society.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. 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*, pages 770–778. IEEE Computer Society.

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

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Alexander Patrick Mathews, Lexing Xie, and Xuming He. 2016. Senticap: Generating image descriptions with sentiments. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 3574–3580. AAAI Press.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. Association for Computational Linguistics.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1556–1566. The Association for Computer Linguistics.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4566–4575. IEEE Computer Society.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3156–3164. IEEE Computer Society.

Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4651–4659. IEEE Computer Society.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Trans. Assoc. Comput. Linguistics*, 2:67–78.

Yiwu Zhong, Liwei Wang, Jianshu Chen, Dong Yu, and Yin Li. 2020. Comprehensive image captioning via scene graph decomposition. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XIV*, volume 12359 of *Lecture Notes in Computer Science*, pages 211–229. Springer.

Grounding Plural Phrases: Countering Evaluation Biases by Individuation

Julia Suter **Letitia Parcalabescu** **Anette Frank**
Department of Computational Linguistics
Heidelberg University, 69120 Heidelberg
{suter,parcalabescu,frank}@cl.uni-heidelberg.de

Abstract

Phrase grounding (PG) is a multimodal task that grounds language in images. PG systems are evaluated on well-known benchmarks, using *Intersection over Union (IoU)* as evaluation metric. This work highlights a disconcerting bias in the evaluation of grounded *plural phrases*, which arises from representing *sets of objects* as a *union box* covering all component bounding boxes, in conjunction with the IoU metric. We detect, analyze and quantify an *evaluation bias* in the grounding of plural phrases and define a *novel metric, c-IoU,* based on a union box's component boxes. We experimentally show that our new metric greatly alleviates this bias and recommend using it for fairer evaluation of plural phrases in PG tasks.

1 Introduction

Phrase grounding (PG) describes the multimodal task of identifying objects in images and connecting them to free-form phrases in a textual description (caption). A phrase usually describes one, or sometimes several, specific objects.

Grounding phrases in image regions provides an essential link between texts and images and serves as a foundation for multimodal understanding tasks, including sentence-to-image alignment, Visual QA, Visual Common-sense Reasoning (VCR), etc.

Benchmarks for training and evaluating PG systems (Everingham et al., 2010; Lin et al., 2014; Kazemzadeh et al., 2014; Plummer et al., 2015; Krishna et al., 2017) generally provide rectangular bounding boxes as ground truth (GT). Therefore a PG ground truth is represented as a phrase linking to a (gold) bounding box enclosing the image patch referred to by the phrase. Some datasets provide pixel segmentation masks (Lin et al., 2014), which enable more precise evaluations but are more difficult and costly to produce. Thus, the trend towards annotating bounding boxes persists in recent datasets (Ilinykh et al., 2019).

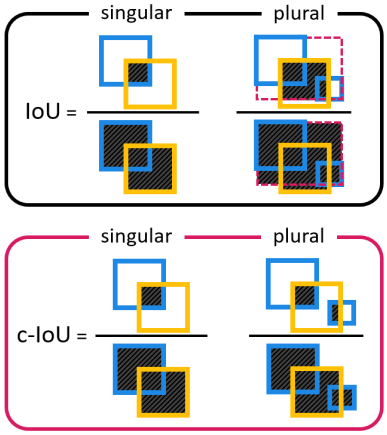


Figure 1: Illustration of how to compute the evaluation metric *IoU* and its adaptation *c-IoU* (with ground truth (GT) bounding boxes in blue, and predicted boxes in yellow). The numerator represents the computed intersection area, the denominator represents the union area. *IoU* and *c-IoU* only differ for plural phrases: *IoU* computes a union box (dashed) covering all components, while *c-IoU only considers* the area of the individual components to compute the intersection and union.

Plural phrases describe multiple entities in an image, either through a collective term (e.g. *crowd*) or a plural form (e.g. *two children*). Depending on the annotation, the gold box consists either of a single box enclosing all entities or several component boxes representing the individual entities. By convention[1], component boxes are merged into one *union box* spanning all individual boxes, functioning as a single gold box. Figure (2.a) gives an example of a union box for a plural phrase with two components. This reduction of multiple boxes to a single union box is widely established in PG evaluation, both for ground truths and predictions.

Although plural phrases are underrepresented in PG benchmarks, they constitute substantial proportions, and appropriate annotation and evaluation of component boxes is essential to achieve high-

[1]as, e.g., adopted in Plummer et al. (2015) and since then presumably adopted in the community for comparability

Proceedings of the Second Workshop on Advances in Language and Vision Research, pages 22–28

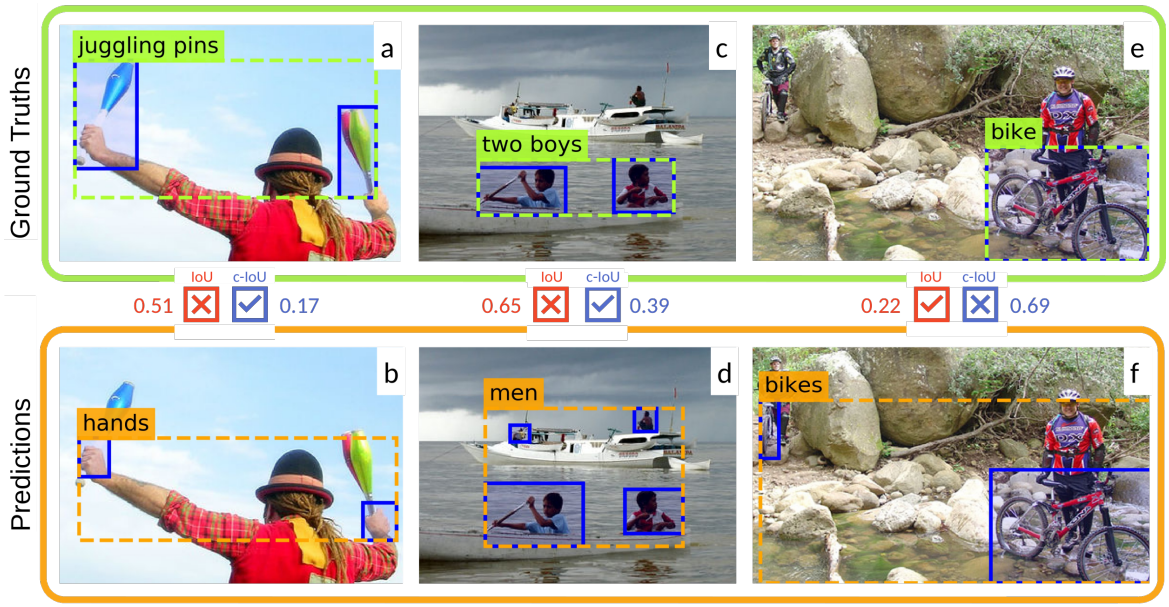


Figure 2: Ground truth (green/top) and prediction (orange/bottom) cases with components (blue) and union boxes (dashed). The Ground Truth box label in green represents the phrase being grounded; the orange phrase marked in the prediction represents the concept fitting the detected plural object. The scores represent the IoU score (red, left) and the c-IoU score (blue, right), respectively. Predictions with scores ≥ 0.5 are considered correct. The check boxes show whether the metrics correctly evaluate the prediction. For example, for sub-figures (a+b) the c-IoU score is 0.17 and so the prediction is considered incorrect (≤ 0.5), thus the c-IoU metric correctly evaluates that the detected *hands* constitute an incorrect prediction for *juggling pins* (blue tick).

quality mappings for all phrase types. However, the annotation of plural phrases is challenging, as shown in Testoni et al. (2020); Marín et al. (2020) who investigate how phrases can refer to groups of objects or several entities within a group.

(Semi-)supervised PG systems generally do not differentiate singular and plural phrases and always predict a single box (Li et al., 2019; Lu et al., 2020; Plummer et al., 2015). For multiple boxes with the same predicted label, either the largest box or the union box is returned. Thus, components are not individually evaluated, and the same metric, *Intersection over Union (IoU)*, can be uniformly applied to a prediction box for any phrase type. IoU computes the ratio of the *area of overlap* over the *area of union* between a predicted and a gold box and is usually thresholded at IoU ≥ 0.5.

While IoU is a simple and effective metric for evaluating 1:1 mappings, we claim that it is unsuitable for evaluating plural phrases. We show that the union box is in fact not an ideal gold representation for plural phrases: it can make the gold box overly large, especially when including areas that do not represent any components, and thus introduces an evaluation bias favoring large prediction boxes. Our contributions are as follows:

i) We detect, describe and *quantify an evaluation bias* in the grounding of plural phrases when applying standard practice of measuring *IoU* over *union boxes*, using an unsupervised PG system on the PG dataset Flickr30k.
ii) We propose a *novel evaluation metric* based on component boxes rather than union boxes.
iii) We show that the *new metric alleviates this bias* and reduces the evaluation failures.

2 Evaluation Bias

IoU is the standard evaluation metric used in PG and rewards predictions that highly overlap with their gold boxes. For a *plural phrase* that links to multiple ground truth boxes, a *union box* enclosing all components is generated, so the same evaluation metric can be used for singular and plural phrase types. However, we argue that this method introduces a considerable bias, which may result in unfair evaluations. When evaluating on union boxes, we ignore all information about the components' sizes and positions, and only consider the union box outline. If components are spread across the image, a union box can become much larger than the combined size of its component boxes, which makes them imprecise and ambiguous.

Figures (2.a) vs. (2.b) show an example of two-component union boxes that are highly overlapping – one for *pins*, the other for *hands*. Hence, a system that returns the prediction (2.b) for *juggling pins* will be unduly considered as correct. Similarly, for a prediction with too few or too many components, IoU often fails to detect such mistakes, as in (2.c) vs. (2.d). The ground truth (2.c) for *two boys* includes only two components, yet a system predicting four components (including the men on the boat) will still be correct according to IoU.

This type of 'false positive' arises from the generation of *union boxes*, in conjunction with the relatively forgiving nature of the IoU metric for large predicted boxes. Given such undesirable failure cases, we conjecture that IoU can lead to unwanted evaluation biases and we investigate whether it is a sound metric for evaluating plural phrases.

2.1 Quantifying the Bias

We verify and quantify the potential evaluation bias on GT annotations of Flickr30k and empirical system predictions. Depending on the distribution of component boxes across the image, the union box can be large, even if the components themselves are small. In Fig. (2.a) 75.47% of the union box area does not represent any component, so we term this area *filler space*. On the complete Flickr30k data, we compute an average of 3.6 components per plural phrase, which on average cover only 68% of the union box area, leaving one third (32%) of the space unfilled. For 24% of all union boxes, the filler space covers more than 50%, the gold box being twice as large as its components.

Hence, there is considerable potential for an evaluation bias to arise, as the IoU metric may unfairly favor large prediction boxes in two ways: i) overly large union boxes allow the prediction of wrong object types that happen to fall into the gold union box area; and ii) even if objects of the correct type are predicted, a large union box may be filled with too many or too few objects compared to the GT, and may still satisfy IoU $\geq$0.5. To verify this hypothesis, we perform experiments using an unsupervised PG system, capable of processing plural phrases.

2.2 Bias in Context of System Performance

Most existing PG systems are (semi-)supervised learners and need to be adapted to the special case of plural phrases: their object detectors need to deliver union boxes, instead of single-object boxes. Since plural phrases are much less frequent than singular phrases, this distributional bias may lead to poorer predictions for plural phrases. Recently, unsupervised PG systems have been proposed (Wang and Specia, 2019; Parcalabescu and Frank, 2020) that achieve competitive performance, but are not subject to such frequency biases. We thus perform our experiments with a system that replicates Wang and Specia (2019)'s approach.

The system[2] maps phrases to predicted bounding boxes using similarity rankings derived from word embeddings for the phrase and the candidate box labels. Since our object detector only detects single objects, we automatically generate plural objects that include several objects, by combining boxes with the same label.

We apply the system to a test set of 10k images with 5 captions each, containing 3.3 phrases on average. We ground ca. 141k phrases, including ca. 31k plural phrases (21.8%) and measure accuracy for the predicted bounding box(es) for a given phrase, using the IoU evaluation metric (with a threshold at 0.5) as success criterion.

Table (1.a) displays evaluation results for *all phrases* vs. *plural phrases* only, and in both cases we distinguish predicted boxes comprising *single* objects only vs. *all* objects (single and plural), for various settings: i) *upper bound* (row 1); ii) performance of our PG system in different settings (rows 2-4); and iii) manipulated predictions, i.e. *max box* and *random predictions* (rows 5-6).

i) Upper bound *Upper bound* represents the highest possible PG performance, computed as percentage of phrases with at least one detected object that matches the GT. Using only *single* objects, we find an upper bound of 72.34 for all phrases and 46.67 for plural phrases. The fact that single objects – which cannot constitute correct groundings for plural phrases – provide 'successful candidates' for nearly half the plural phrases, emphasizes that IoU is not a suitable metric for plural phrase grounding. When considering boxes with multiple objects as candidates, the upper bound increases by 2.82 percentage points (pp.) to 75.16 on all phrases and by 20.98 pp. to 67.65 on plural phrases, demonstrating that *plural objects* are an essential addition.

ii) PG system evaluation This setting also shows an increase when considering *plural objects*, with an increase for *all phrases* by 1.69 pp., and for *plural phrases* by 5.45 pp. Candidate pruning fur-

[2]Details of the system are given in the Appendix.

a) IoU metric	All Phrases		Plural Phrases	
	single	*all*	*single*	*all*
Upper bound [+prun.]	72.34	75.16	46.67	67.65
Unsupervised PG	47.94	49.63	31.15	36.60
– [+pruning]	-	53.36	-	56.46
– [+pruning, +enlarged]	47.99	52.03	33.84	52.45
Max box predictions	23.63	23.63	32.19	32.19
Random predictions	17.97	20.75	24.09	29.17

b) c-IoU metric	All Phrases		Plural Phrases	
	single	*all*	*single*	*all*
Upper bound [+prun.]	72.34	73.69	46.69	60.94
Unsupervised PG	48.05	49.94	31.00	37.37
– [+pruning]	-	52.38	-	50.09
– [+pruning, +enlarged]	47.26	51.02	29.84	46.95
Max box predictions	21.45	20.98	22.88	22.19
Random predictions	9.17	13.86	7.62	14.08

Table 1: PG performance in accuracy computed with **IoU** vs. **c-IoU** on *all* vs. *plural phrases*, considering *single* object boxes vs. *all* (single & multiple) object boxes. *+pruning* filters candidates: for plural phrases we consider plural objects only; for singular phrases only single objects. *+enlarged*: size of detected objs. increased by 50%.

ther increases accuracy by 3.73 pp. on *all phrases* and 19.86 pp. on *plural phrases*, while limiting the potential exploitation of large candidate boxes.

iii) Manipulated predictions We hypothesized that using IoU with union boxes is too forgiving and favors large predictions, so we conduct experiments where we generate overly large object predictions: in one, predictions cover the entire image (*max box predictions*); in the other, original predictions are enlarged by 50%. For the *max box predictions*, we obtain overall 23% correct predictions, and 32% for *plural phrases*. Thus, every third plural phrase benefits from very large prediction boxes. Ideally, IoU is designed to punish predictions that are overly large or not well placed over the gold box, due to division by union area. However, the image frame limits the maximum box size to the size of the image, which reduces the normalization effect for large objects.

Measuring PG performance with enlarged prediction boxes [+enlarged], increases accuracy by 2.69 pp. for *plural phrases* when considering singular objects only – despite singular objects being unsuitable predictions by definition. This further supports our hypothesis that large predictions are generally favored. However, larger predictions do not increase performance on mixed phrase types, so singular phrases must be less affected by this bias. For plural objects, PG performance even decreases with enlarged predictions, which suggests that plural objects cannot benefit from expansion.

In sum, the high upper bound with singular objects for plural phrases, the strong performance when predicting the entire image, and the effect on PG performance when enlarging prediction boxes all support our hypothesis that *the evaluation of plural phrases by IoU is biased*. Hence, a new metric is needed to counter this bias.

3 Our new Evaluation Metric c-IoU

We aim at a metric that is not based on the union box, but its components. However – any metric is only as good as the quality of its underlying ground truth. When studying the annotation of plural phrases in Flickr30k, we found that many of them are imprecise or incomplete. Nearly one third of plural phrases are annotated with a single bounding box without components and for 9% the number of components does not match the cardinality of the referring phrase (e.g. two component boxes for *three women*), leaving 37% of the plural phrases without proper representation of their components. This high level of noise precludes any metric that relies on matching the number of component boxes of ground truth and predictions.

In §2.1 we identified the filler space of union boxes – jointly with IoU evaluation – as the source of the detected evaluation bias. To combat this, we define an *adapted IoU* that is not computed over the union box, but its *aggregated components*, by taking the intersection of all gold and predicted component boxes and dividing by the area of the union of all (gold and predicted) component boxes. We call this metric *component IoU (**c-IoU**)*. c-IoU is analogous to standard IoU for single-object boxes, and only affects the evaluation of plural phrases, as seen in Fig. (1). By considering the area covered by *all* component boxes, it gains robustness against annotation noise.

Fig. (2.a-d) show two examples where IoU fails to correctly evaluate predictions, while c-IoU succeeds. The prediction *hands* for the phrase *juggling pins* yields an IoU score of 0.51, which accepts the prediction. The c-IoU score of 0.17, by contrast, rejects this prediction. Similarly, the prediction *men* is considered correct for *two boys* by IoU (0.65), but correctly rejected by c-IoU (0.39).

4 Evaluation of Component IoU (c-IoU)

We evaluate **c-IoU** in the same way as we did in §2.2 to quantify biases under IoU, and give results in Table (1.b). We expect c-IoU to avoid biases for plurals and ensuing false predictions.

The experiments confirm our expectation: large prediction boxes yield lower scores with c-IoU. *Max box predictions*, computed on all objects found in the image, yields 9.31 pp. lower accuracy on plural phrases. PG system performance with enlarged predictions measured on singular objects for plural phrases increases by 2.69 pp. to 33.84 for IoU, yet decreases by 1.16 pp. to 29.84 for c-IoU. Therefore, c-IoU better detects wrong predictions.

But c-IoU does not catch all incorrect predictions: with pruning, accuracy measured with c-IoU increases less (+12.72 pp. to 50.09) compared to IoU (+19.86 pp. to 56.46). Data inspection shows that without pruning, c-IoU allows plural objects for singular phrases (and vice versa) – typically the plural object includes the targeted object plus another small object in the background (see Fig. (2.e+f)). Since the background object drastically expands the union box but not the component union area, IoU may correctly evaluate such cases, while c-IoU could fail. Hence, a combination of both metrics could be beneficial, where c-IoU ensures that the right components are selected, while IoU may detect out-of-focus objects.

As a final test, we evaluate both metrics on artificially generated false plural box predictions. For each phrase, we assemble a *random prediction* consisting of 2-5 components with different labels. Ideally, most predictions should be labeled as incorrect, thus a lower accuracy indicates a more sensitive metric. c-IoU indeed returns much lower scores than IoU: 9.17 and 7.62 (c-IoU) vs. 19.97 and 24.09 (IoU) on *all phrases* and *plural phrases*, showing that c-IoU effectively counters the bias.

5 Conclusion

We have detected, described and quantified an evaluation bias for plural phrases in the PG literature. Our alternative c-IoU metric, acting on components rather than union boxes, alleviates this bias, as we show in experiments with an unsupervised PG system. Future work could test more systems to assess by how much state-of-the-art performance is lower than currently estimated. Evaluation of plural phrases is further impeded by the low quality of the gold boxes. Therefore, future benchmarks need to annotate plural phrases with all their components if we wish to enable PG systems to better learn the intricacies of language (including plural expressions) in relation to the visual modality.

References

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338.

Nikolai Ilinykh, Sina Zarrieß, and David Schlangen. 2019. Tell me more: A dataset of visual scene description sequences. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 152–157.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. ReferItGame: Referring to Objects in Photographs of Natural Scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual Genome: Connecting Language and Vision using Crowdsourced Dense Image Annotations. *International journal of computer vision*, 123(1):32–73.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. VisualBERT: A Simple and Performant Baseline for Vision and Language. *arXiv preprint arXiv:1908.03557*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 2020. 12-in-1: Multi-task vision and language representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10437–10446.

Nicolás Marín, Gustavo Rivas-Gervilla, and Daniel Sánchez. 2020. A preliminary approach to referring to groups of objects in images. In *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–7. IEEE.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Letitia Parcalabescu and Anette Frank. 2020. Exploring Phrase Grounding Without Training: Contextualisation and Extension to Text-Based Image Retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 962–963.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k Entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649.

Alberto Testoni, Claudio Greco, Tobias Bianchi, Mauricio Mazuecos, Agata Marcante, Luciana Benotti, and Raffaella Bernardi. 2020. They are not all alike: Answering different spatial questions requires different grounding strategies. In *Proceedings of the Third International Workshop on Spatial Language Understanding*, pages 29–38.

Josiah Wang and Lucia Specia. 2019. Phrase Localization Without Paired Training Examples. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4663–4672.

A Appendix

System details Our approach replicates the unsupervised *bag-of-objects approach* by Wang and Specia (2019). The phrase and the labels for candidate objects are embedded using 300-dimensional word2vec embeddings (Mikolov et al., 2013). The object candidates are ranked by their *cosine similarity* to the phrase, and the object with the most similar label is returned. If there are several objects for the highest ranking label, we return the largest one in case of singular phrases and the union box (plural object) in case of plural phrases. In contrast to prior systems that used (multiple) object detectors with large label sets (545 or 1600 labels), our object detector, trained on Visual Genome (Krishna et al., 2017), uses only 150 coarse-grained labels.

We test performance on the Flickr30k Entities (Plummer et al., 2015) dataset for phrase grounding. The test set consists of 10k images with 5 captions each, containing 3.3 phrases on average. We ground 140 972 phrases, including 30 762 plural phrases (21.8%). The vocabulary of the phrases is relatively diverse with 8301 different words on the test split.

Evaluation examples Fig. (3) shows a few more examples of ground truths and our system's predictions, as well as the correctness of the evaluation using IoU and c-IoU. Fig. (3.a-d) show examples where IoU accepts predictions with incorrect object labels, while c-IoU rejects them. In (3.e+f), c-IoU finds that two hats are missing while IoU accepts the incomplete prediction. For example (3.g+h), c-IoU fails to identify that the prediction has a missing component. IoU correctly evaluates the prediction as incorrect because of the obvious union box difference, which makes the IoU drop below 0.5. Example (3.i+j) is a challenging case, as the phrase [*two young men clutch rags in]* ***their hands*** requires context for correct grounding, which is not provided by a PG system that looks at phrases individually. As expected, our system additionally predicts the old man's hand, which is incorrect but since the superfluous hand has a small area and is located closely to the others, both evaluation metrics fail to detect this mistake. Finally, in (3.k+l) the ground truth is missing the annotation of the components, so that c-IoU cannot correctly evaluate this correct prediction.

Figure 3: GT and prediction cases with union boxes (dashed) and components (blue); the check marks show whether IoU (red) and c-IoU (blue) correctly evaluate the prediction (for further explanation see Figure 2).

PanGEA: The Panoramic Graph Environment Annotation Toolkit

Alexander Ku* **Peter Anderson*** **Jordi Pont-Tuset** **Jason Baldridge**
Google Research
{alexku, pjand, jponttuset, jridge}@google.com

Abstract

PanGEA, the Panoramic Graph Environment Annotation toolkit, is a lightweight toolkit for collecting speech and text annotations in photo-realistic 3D environments. PanGEA immerses annotators in a web-based simulation and allows them to move around easily as they speak and/or listen. It includes database and cloud storage integration, plus utilities for automatically aligning recorded speech with manual transcriptions and the virtual pose of the annotators. Out of the box, PanGEA supports two tasks – collecting navigation instructions and navigation instruction following – and it could be easily adapted for annotating walking tours, finding and labeling landmarks or objects, and similar tasks. We share best practices learned from using PanGEA in a 20,000 hour annotation effort to collect the Room-Across-Room dataset. We hope that our open-source annotation toolkit and insights will both expedite future data collection efforts and spur innovation on the kinds of grounded language tasks such environments can support.

1 Introduction

The release of high-quality 3D building and street captures (Chang et al., 2017; Mirowski et al., 2019; Mehta et al., 2020; Xia et al., 2018; Straub et al., 2019) has galvanized interest in developing embodied navigation agents that can operate in complex human environments. Based on these environments, annotations have been collected for a variety of tasks including navigating to a particular class of object (ObjectNav) (Batra et al., 2020), navigating from language instructions aka vision-and-language navigation (VLN) (Anderson et al., 2018b; Chen et al., 2019; Qi et al., 2020; Ku et al., 2020), and vision-and-dialog navigation (Thomason et al., 2020; Hahn et al., 2020). To date, most of these data collection efforts have required the development of custom annotation tools.

To expedite future data collection efforts, in this paper we introduce PanGEA, an open-sourced annotation toolkit designed for these settings.[1] Specifically, PanGEA assumes an environment represented by discrete navigation graphs connecting high-resolution 360° panoramas, where each node represents a unique viewpoint in the environment and actions involve moving between these viewpoints. Examples of suitable environments include the indoor buildings from Matterport3D (Chang et al., 2017) (using the navigation graphs from Anderson et al. (2018b)) and the street-level environments from StreetLearn (Mirowski et al., 2019).

Out of the box, PanGEA supports two annotation modes: the *Guide* task and the *Follower* task. In the Guide task, Guides look around and move through an environment to follow a pre-defined path and attempt to create a navigation instruction for others to follow. In the Follower task, annotators listen to a Guide's instructions and attempt to follow the path. These annotation modes are based on the Vision-and-Language Navigation (VLN) setting proposed by Anderson et al. (2018b). However, compared to similar annotation tools, PanGEA includes substantial additional capabilities, notably:

- annotation via voice recording (in addition to text entry)
- virtual pose tracking to record what annotators look at
- utilities for aligning a transcript of the words heard or uttered by each annotator with their visual perceptions and actions
- integration with cloud database and storage platforms
- a modular API facilitating easy extension to new tasks and new environments

PanGEA has already been used in two papers. It was used to collect Room-Across-Room (RxR) (Ku et al., 2020), a dataset of human-annotated navigation instructions in English, Hindi and Telugu

*First two authors contributed equally.

[1] github.com/google-research/pangea

Proceedings of the Second Workshop on Advances in Language and Vision Research, pages 29–33

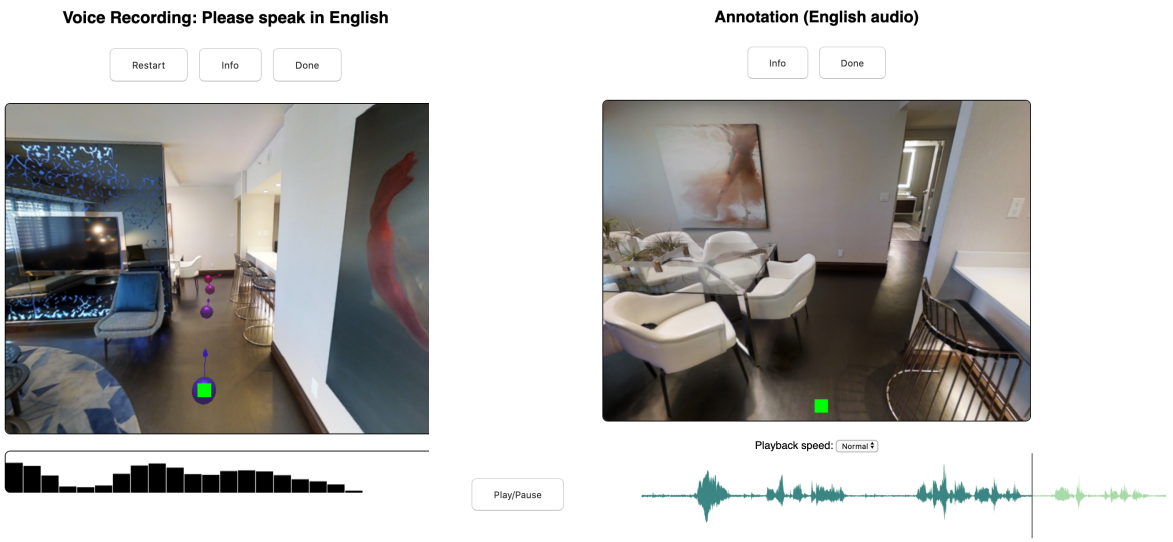


Figure 1: Screenshots of the PanGEA Guide and Follower interfaces. In the Guide task (left), Guides explore a given path while attempting to create a navigation instruction for others to follow. Guides can pause and restart the audio recording at any time. After recording is completed, Guides transcribe their own audio. In the Follower task (right), annotators listen to a Guide's instructions and attempt to follow the intended path. Followers can skip around the Guide's audio using the audio waveform at bottom right. In both tasks, PanGEA tracks the annotators virtual camera pose and automatically aligns it with the Guide's audio transcript.

which is the largest VLN dataset by an order of magnitude. PanGEA was also used to perform human evaluations of model-generated navigation instructions in Zhao et al. (2021). It could be trivially adapted to other tasks that combine annotation with movement, such as annotating walking tours, or finding and labeling particular landmarks or objects.

We next describe PanGEA's capabitilities in more detail. In the final section we share some best practices learned from using PanGEA to collect RxR, which required more than 20,000 annotation hours.

2 PanGEA Toolkit

Guide Task In the Guide task (Figure 1, left), Guides look around and move to explore an environment while recording an audio narration. For the RxR data collection, the Guide's movement was restricted to a particular path through the environment, and annotators were instructed to record navigation instructions that would be sufficiently descriptive for others to follow the same path. However, this restriction can be relaxed to allow free movement and narration for other purposes. Once the Guide is satisfied with their recording, they are asked to manually transcribe their own voice recording into text. This ensures high quality transcription results.

During the Guide task, in parallel to the annotator's voice recording, PanGEA captures a timestamped record of the annotator's virtual camera movements, which we call a *pose trace*. By default, PanGEA is configured to use Firebase[2], saving the Guide's audio recording to a cloud storage bucket, and the transcript, pose trace and other metadata to a cloud database for post processing. Inspired by Localized Narratives (Pont-Tuset et al., 2020), PanGEA includes a utility to automatically align each Guide's pose trace with the manual transcript of their audio recording. This is achieved by using a Speech to Text service[3] to first generate a noisy-but-timestamped automatic transcription. PanGEA then using dynamic time warping to align tokens in the automatic transcript to the manual transcript before propagating timestamps from the automatic to the manual transcription (Figure 2). The result is fine-grained synchronization between the transcribed text, the pixels seen, and the actions taken by the Guide.

Follower Task In the Follower task (Figure 1, right), Followers begin at a specified starting point in an environment and are asked to follow a Guide's

[2] https://firebase.google.com

[3] https://cloud.google.com/speech-to-text

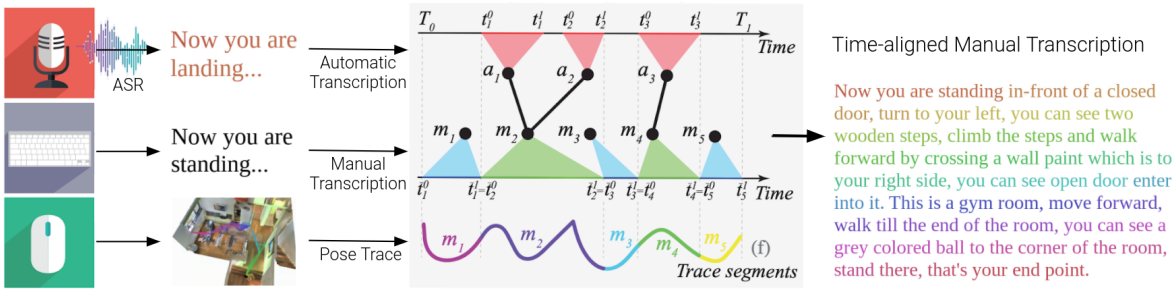


Figure 2: PanGEA time-aligns each annotators manual audio transcription (middle) to a *pose trace* recording their virtual camera movements (bottom). This is achieved by first generating a noisy-but-timestamped automatic transcription (top), which is aligned with the manual transcription using dynamic time warping in order to propagate timestamps to the manual transcription. Figure adapted from Pont-Tuset et al. (2020)

instructions. They observe the environment and navigate as the Guide's audio plays. Followers can skip forward or backward in the audio recording by clicking on an audio waveform representation of the Guide's recording. This allows them to skip over periods of silence or to listen to part of the audio again. Once the Follower believes they have reached the the end of the path, or they give up, they indicate they are done and the task ends. Note that although the Follower task supports audio instructions, it can be easily adapted to replace the audio instruction with a textual instruction. This was the approach taken by Zhao et al. (2021).

As with the Guide task, the Follower's pose trace is recorded and saved to a cloud database, along with the timestamp of the Guide's audio that the Follower listened to at each moment. This allows the Follower's visual percepts and actions to be accurately aligned with text tokens in the Guide's instructions. Similarity between the annotated (Guide) path and the Follower path is also a natural measure of the joint quality of both the Guide and the Follower annotations. In the experiments for RxR, the path extracted from the Follower's pose trace was also used as additional supervision when training Follower agents, since it represents a step-by-step account of how a human solved the task and the visual inputs they focused on in order to do so (Ku et al., 2020).

Deployment PanGEA comes with several demos using a very simplistic environment. To deploy PanGEA for a new large-scale collection effort requires completing 3 main steps:

- Creating a new app in Firebase to initialize the cloud storage and cloud database,
- Setting up an appropriate crowdsourcing platform to serve the PanGEA front-end to a pool of annotators, and
- Setting up the environment to be used, e.g., hosting the images and navigation graphs in a storage bucket in an appropriate format.

Further details are provided in the PanGEA readme.

3 Observations and Best Practices

PanGEA was developed for the collection of the RxR dataset, a 20,000+ hour annotation effort based on Matterport3D indoor scenes. Many of the lessons learned during this collection effort are codified in the PanGEA toolkit. For example, we found that uploading recorded audio at the end of the Guide task was time consuming, and so in the final version of PanGEA the wav file is uploaded in the background while the annotator is busy transcribing their audio. We also found that audio annotations could include long periods of silence, so we provided Follower annotators with an audio waveform visualization and an interface to skip over silence. Some other observations and best practices for reducing annotation times and improving annotation quality are shared in this section.

Annotators Complete Tasks in Creative Ways PanGEA is designed to capture the alignment between annotators' visual percepts, actions and utterances to provide fine-grained spatio-temporal grounding. In initial trials with PanGEA, we found that some annotators – with the best of intentions – completely undermined this paradigm. We had envisioned them speaking while moving and looking at the environment; however, in an effort to generate more fluent instructions, some annotators first explored the environment while drafting a nav-

igation instruction separately in a text editor. Then, having finalized the textual instruction, the annotator read it all at the end of the audio recording. While this strategy indeed produced high-quality navigation instructions, the instructions were no longer time-aligned to the pose trace. Interestingly, the language used in the instructions also differed. Instructions drafted as text tended to use more connective phrases — for example, "turn right *and then* you will see a dining table" instead of "turn right... *now* you see a dining table". We found it challenging to add guardrails in PanGEA that could prevent this behaviour without unduly restricting the freedom of the annotators and the flexibility of the toolkit. Instead, we addressed this issue–successfully–via explicit training.

Annotator Training To overcome the aforementioned issue and to improve annotation quality in general, for RxR, we conducted an interactive virtual training session with annotators, providing examples of ideal annotations and various failure modes. Annotators were also able to ask questions regarding how to best complete the tasks assigned to them. Although interactive training sessions are not always possible, at minimum we recommend providing annotators with a training video that shows a walk-through of the task and notes common pitfalls to avoid. We provide links to the demo videos for the RxR Guide task[4] and Follower task[5] (initially called the Tourist task).

Pilot Collections and Learning Periods Annotating and following navigation instructions in a virtual world is a complex task. We recommend allowing for several small-scale pilot data collections to identify issues with the collection process. This includes having the team creating the dataset perform the tasks using the tool. Secondly, we recommend allowing for a learning period whenever a new annotator is introduced to the task, i.e., planning to discard the first 5–10 annotations produced by a new annotator. We found that rotating annotators between both Guide and Follower tasks early in their experience improved annotation quality because doing so provides much greater awareness of the needs of Followers when completing the Guide tasks.

Data Monitoring Dashboard We recommend using VLN evaluation metrics such as success rate, navigation error and SPL (Anderson et al., 2018a) (or similar metrics for alternative tasks) to continually monitor the quality of the collected Guide navigation instructions and Follower paths. By storing the collected annotations in Firebase, it is relatively easy to construct web-based interfaces to monitor these metrics. In the case of RxR, we created a monitoring dashboard that displayed success rates for each annotator pool and also each annotator, with the capability to replay the pose traces from individual Guide and Follower annotations. Annotators were able to see an anonymized view on their progress as it related to others, which helped them assess whether they were performing the task correctly or needed additional changes and perhaps explicit guidance.

Speech versus Writing In tasks that require a person to perform actions while producing or comprehending language, it is much easier if people are allowed to use speech rather than writing because it allows them to use their hands and eyes fully for performing actions. This has very real consequences for thinking about future data collection efforts. Speech interactions will be essential for any tasks that include time pressure, such as collaborative games where players use language to coordinate. There is also a simple but significant cost advantage: on average, the transcription portion for an RxR Guide annotation took three to four times longer than collecting the speech instruction itself, so either a great deal more instructions could have been collected, or the cost could have been significantly reduced. Speech also encodes intonation and is more likely to elicit interesting dialectal differences. For these and other reasons, we may thus want to encourage more research that works on language grounding tasks that work with speech directly, and provide current best automatic speech recognition output for those who insist on working with text only.

4 Future Applications

There are many potential future applications of PanGEA and tools that could be built based on the design decisions discussed above. We are particularly excited about multi-agent problems that collect pose traces from multiple participants as they coordinate via language, such as hide and seek games or tasks where items must be moved from one location to another to satisfy goals or solve puzzles, similar to CerealBar (Suhr et al., 2019).

[4] https://youtu.be/aJkJfB8oI2M
[5] https://youtu.be/vcP-oX1t0CU

References

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. 2018a. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. 2018b. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, pages 3674–3683.

Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. 2020. Objectnav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171*.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3d: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*.

Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. 2019. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *CVPR*.

Meera Hahn, Jacob Krantz, Dhruv Batra, Devi Parikh, James M. Rehg, Stefan Lee, and Peter Anderson. 2020. Where are you? localization from embodied dialog.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. 2020. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. *EMNLP*.

Harsh Mehta, Yoav Artzi, Jason Baldridge, Eugene Ie, and Piotr Mirowski. 2020. Retouchdown: Adding touchdown to streetlearn as a shareable resource for language grounding tasks in street view. *EMNLP Workshop on Spatial Language Understanding (SpLU)*.

Piotr Mirowski, Andras Banki-Horvath, Keith Anderson, Denis Teplyashin, Karl Moritz Hermann, Mateusz Malinowski, Matthew Koichi Grimes, Karen Simonyan, Koray Kavukcuoglu, Andrew Zisserman, et al. 2019. The streetlearn environment and dataset. *arXiv preprint arXiv:1903.01292*.

Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. 2020. Connecting vision and language with localized narratives. In *ECCV*.

Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. 2020. Reverie: Remote embodied visual referring expression in real indoor environments. In *CVPR*.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. 2019. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*.

Alane Suhr, Claudia Yan, Jack Schluger, Stanley Yu, Hadi Khader, Marwa Mouallem, Iris Zhang, and Yoav Artzi. 2019. Executing instructions in situated collaborative interactions. 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)*, pages 2119–2130, Hong Kong, China. Association for Computational Linguistics.

Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. 2020. Vision-and-dialog navigation. In *Conference on Robot Learning (CoRL)*, pages 394–406. PMLR.

Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. 2018. Gibson env: real-world perception for embodied agents. In *CVPR*. IEEE.

Ming Zhao, Peter Anderson, Vihan Jain, Su Wang, Alex Ku, Jason Baldridge, and Eugene Ie. 2021. On the evaluation of vision-and-language navigation instructions. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Learning to Learn Semantic Factors in Heterogeneous Image Classification

Boyue Fan
University of Sheffield
United Kingdom

Zhenting Liu
Pasadena City college
United States

Abstract

Few-shot learning is to recognize novel classes with a few labeled samples per class. Although numerous meta-learning methods have made significant progress, they struggle to directly address the heterogeneity of training and evaluating task distributions, resulting in the domain shift problem when transitioning to new tasks with disjoint spaces. In this paper, we propose a novel method to deal with the heterogeneity. Specifically, by simulating class-difference domain shift during the meta-train phase, a bilevel optimization procedure is applied to learn a transferable representation space that can rapidly adapt to heterogeneous tasks. Experiments demonstrate the effectiveness of our proposed method.

1 Introduction

Deep learning methods are now widely used in diverse applications. However, their efficacy is largely contingent on a large amount of labelled data in the target task and domain of interest (Vaswani et al., 2017). Different from humans that can easily learn to accomplish new tasks with a few examples, it is difficult for machines to rapidly generalize to new concepts with very little supervision, which calls considerable attention to the challenging few-shot learning (FSL) setting. For example, few-shot classification problem requires models to classify unlabeled samples into novel classes with only a few labeled samples available for training (Finn et al., 2017). Commonly understood as learning to learn, meta-learning paradigm has made significant progress in FSL by transferring knowledge extracted from a collection of previous tasks (Vinyals et al., 2016; Snell et al., 2017). Such task-agnostic knowledge can contribute to the current testing task with optimizing learning algorithms. However, beyond its recent achievements, meta-learning still faces the problem of generalization.

In contrast to supervised machine learning methods which assume that training and testing data are sampled i.i.d. from the same distribution, FSL aims to learn to address tasks from different distributions with limited data. This refers to the realistic scenario that the label spaces of future testing tasks can not be obtained in advance and are often disjoint with the label spaces of training tasks. In experiments, this is actualized by splitting all categories in the dataset into non-overlapping base classes and novel classes, while training tasks are sampled from base classes and testing tasks are samples from novel classes. Therefore, due to the class label difference, meta-learning approaches suffer from natural heterogeneous distributions of tasks. As each task can be regarded as having a separate domain, it can be considered as a special case of domain shift that is extremely serious when a large gap of semantic relationship exists between base classes and novel classes.

As most of the current meta-learning approaches make a strong assumption that training tasks and testing tasks are drawn from the similar distributions and share the same characteristics, (Chen et al., 2019) has shown the limitations of existing approaches in cross-domain FSL scenarios where base classes and novel classes are from different datasets. However, few works have focused on this issue to improve existing approaches. For example, as a representative work of metric-based meta-learning, Prototypical Network (Snell et al., 2017) learns a metric space where embeddings of query samples in one class are close to the centroid of support samples in the same class, and far from centroids of other classes in the task. While Prototypical Network benefits from a simple but effective inductive bias, it lacks adaptation to new tasks or domains.

In this paper, we propose to improve such metric-based approaches with a bilevel optimization procedure. Specifically, we simulate class-difference-caused domain shift during meta-training by simultaneously sampling multiple tasks with non-

Proceedings of the Second Workshop on Advances in Language and Vision Research, pages 34–38

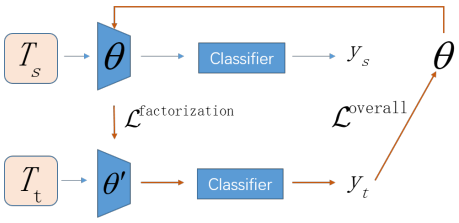


Figure 1: Overview of our proposed Meta-ProtoNet.

overlapping class sets. Each time one of the tasks is prepared as the target task for outer level optimization and the others are first used as the source tasks for inner level optimization of the network. Following this training strategy during the meta-train phase, the model can better adapt to the testing tasks from heterogeneous distributions with an adaptation step.

Moreover, different from some usual options of inner objective, we use Shannon entropy as an unsupervised factorization loss to constrain the learned representations as near-binary codes (Chang et al., 2019). This can be viewed as learning a discriminative latent factor space for each task where each factor can be interpreted as a latent attribute that is corresponding to abstract visual concepts.

To summarize, our main contributions are :1) considering the challenge of heterogeneous task distributions faced by few-shot learning, we simulate the class-difference-caused domain shift in the meta-train phase, and devise a metric-based meta-learning approach integrated with a bilevel optimization for better generalization; 2) we propose to utilize an unsupervised factorization loss as the inner objective, making representations to be near-binary codes that reduce the difficulty of classifier learning. Meanwhile, due to the bilevel optimization between heterogeneous few-shot tasks during meta-training, the model can rapidly learn the representation space for testing tasks; 3) We conduct extensive experiments and analysis to demonstrate that our approach effectively improves the performance and interpretability under both conventional and cross-domain few-shot settings without introducing additional architectures, and thus it can be regarded as a better baseline.

2 Methodology

2.1 Prototypical Network.

As a simple but effective model for FSL learning, Prototypical Network (ProtoNet) (Snell et al., 2017) use an embedding function f_θ with parameters θ to encode each sample into a representation vector. For each class c in the class set C of the task T , a prototype vector p_c is defined as the mean vector of the embedded support samples in the class, which can be expressed as $p_c = \frac{1}{|\mathcal{S}_c|} \sum_{(x_i, y_i) \in \mathcal{S}_c} f_\theta(x_i)$. When inferring, the probability over classes for a query sample x_i is a softmax over the inverse of squared Euclidean distances between the query representation and prototype vectors, expressed as $P_\theta(y_i = c \mid x_i) = \frac{\exp\left(-\|f_\theta(x_i) - p_c\|^2\right)}{\sum_{c' \in C} \exp\left(-\|f_\theta(x_i) - p_{c'}\|^2\right)}$. The classification loss is the sum of negative log-probability of each query sample in task T with its ground-truth class label: $\mathcal{L}^{\text{classification}}(\theta) = -\sum_{c \in C} \sum_{x_i \in \mathcal{Q}_c} \log P_\theta(y_i = c \mid x_i)$.

2.2 Learning Latent Factors

As the embedding function f_θ of Prototypical Network can be any deep neural network, it is often organized as a convolutional neural network (CNN) for image classification tasks. In our MetaProtoNet, we set the activation function of the last layer to Sigmoid function $\sigma(x) = \frac{1}{1+\exp(-x)}$ instead of the most commonly used ReLU function. This limits the scale of the learned representations $f_\theta(x_i) \in (0,1)^d$, where d denotes the dimension number of the representations. Deep architectures are capable of learning to extract useful infor mation from the samples, and potentially construct representations as the composition of the local abstract concepts that are useful for downstream tasks. Therefore, Sigmoid activated outputs of f_θ can be viewed as multi-label predictions on latent factors, as the activation of each dimension closer to 0 or 1 can be interpreted as the corresponding visual attributes being present and absent. Moreover, Meta-ProtoNet constrains the learned representations to become near-binary codes by applying Shannon entropy as an unsupervised factorization loss, expressed as

$$\mathcal{L}^{\text{factorization}}(\theta) = -\sum_{x_i \in \{\mathcal{S}, \mathcal{Q}\}} \langle f_\theta(x_i), \log(f_\theta(x_i)) \rangle \tag{1}$$

where $\log(\cdot)$ is applied element-wise, and $\langle \cdot, \cdot \rangle$ denotes the vector inner product operation. This not only encourages the representations to become more interpretable but also decreases the uncertainty of latent factors discovery.

2.3 Training Meta-ProtoNet

According to (Snell et al., 2017), Prototypical Network can be re-interpreted as a linear classifier that is applied to the representations learned by the nonlinear embedding function. With the improvement above, near-binary representations generated by the embedding function are expected to be preferable for the jointly learned linear classifier without sacrificing representation power and differentiable optimization for exactly binary codes (Li et al., 2017). However, it would result in a suboptimal representation space for heterogeneous testing tasks since the metric-based approach is no longer updated to adapt to new domains in the meta-test phase. To overcome the approaching domain shift problem, we devise a bilevel optimization procedure for a fast adaptation to the feature distribution in the new task.

Specifically, instead of randomly sampling a single task, we simultaneously sample m tasks $\mathcal{T}_{\text{set}} = \{\mathcal{T}_1, \cdots, \mathcal{T}_m\}$ without class overlap from the distribution over training tasks $p\left(\mathcal{T}^{tr}\right)$ in the metatrain stage. For each task in $\mathcal{T}_{\text{set}}$, we first denote it as the target task $\mathcal{T}_t$ and obtain a copy of the model parameters θ as θ', then θ' is updated by minimizing the factorization loss over each task $\mathcal{T}_s$ in the source tasks $\mathcal{T}_{\text{set}} - \mathcal{T}_t$. Each update of θ' can be expressed as

$$\theta' = \theta' - \alpha \nabla_{\theta'} \mathcal{L}^{factorization}\left(\theta'\right) \tag{2}$$

where α is the inner learning rate. This is viewed as the inner level of the bilevel optimization procedure, and after all of $\mathcal{T}_s$ are used for the update of θ', we utilize $\mathcal{T}_t$ to optimize the model. Specifically, the model parameters θ are updated as follows:

$$\theta = \theta - \beta \nabla_{\theta} \mathcal{L}^{\text{overall}}\left(\theta'\right) \tag{3}$$

where β is the outer learning rate. The meta-optimization is performed over the model parameters θ, whereas the objective $\mathcal{L}^{\text{overall}}\left(\theta'\right)$ is computed using the updated model parameters θ' and can be expressed as

$$\mathcal{L}^{\text{overall}}\left(\theta'\right) = \mathcal{L}^{\text{classification}}\left(\theta'\right) + \gamma \mathcal{L}^{\text{factorization}}\left(\theta'\right) \tag{4}$$

where γ is the trade-off hyperparameter. The key idea underlying the algorithm is that to alleviate the class-difference-caused domain shift, the task-specific knowledge including semantic information of categories is decomposed into reusable low-level task-agnostic knowledge by transferring latent factors across heterogeneous tasks. Each round of bilevel optimization can be viewed as a simulation of the whole process including meta-train and meta-test: In the inner level (corresponding to the meta-train phase), we encourage the model to learn to generate latent factors for tasks drawn from the source distribution. As high performance of classification on these tasks is not necessary and may be detrimental to the classification of heterogeneous target tasks, the inner objective only aims to discover latent factors and does not include classification loss. Moreover, we expect the learned latent factor space to be transferable, and thus the learning process of the source tasks can promote the learning of heterogeneous tasks. Therefore, in the outer level (corresponding to the meta-test phase), the model is optimized with the overall loss including classification loss and factorization loss.

2.4 Testing Meta-ProtoNet

In the meta-test phase, when adapting to each new testing task $\mathcal{T}_j$, the trained parameters θ are updated to θ' using only one gradient descent step with the factorization loss over $\mathcal{T}_j$. Therefore, a task-specific latent factor space of $\mathcal{T}_j$ is learned. The evaluation metric (i.e., the classification accuracy) is calculated with the updated parameters θ'.

3 Experiments

Datasets. In this paper, we address the few-shot classification problem under both conventional and cross-domain FSL settings. These settings are conducted on three benchmark datasets: miniImageNet (Vinyals et al., 2016), Caltech-UCSD-Birds 200-2011 (CUB) (Wah et al., 2011), and SUN Attribute Database (SUN) (Patterson et al., 2014).

Experimental Settings. We conduct experiments on 5-way 1-shot and 5-way 5-shot settings, there are 15 query samples per class in each task. We report the average accuracy (%) and the corresponding 95% confidence interval over the 2000 tasks randomly sampled from novel classes. To fairly evaluate the original performance of each method, we use the same 4-layer ConvNet (Vinyals et al., 2016) as the backbone for all methods and do not adopt any data augmentation during training. All methods are trained via SGD with Adam (Kingma and Ba, 2014), and the initial learning rate is set to e^{-3}. For each method, models are trained for 40,000 tasks at most, and the best model on the vali-

Method	miniImageNet → CUB		miniImageNet → SUN		CUB → miniImageNet	
	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
Meta-Learner LSTM	23.77	30.58	25.52	32.14	22.58	28.18
MAML	40.29	53.01	46.07	59.08	33.36	41.58
Reptile	24.66	40.86	32.15	50.38	24.56	40.60
Matching Network	38.34	47.64	39.58	53.20	26.23	32.90
Prototypical Network	36.60	54.36	46.31	66.21	29.22	38.73
Relation Network	39.33	50.64	44.55	61.45	28.64	38.01
Baseline	24.16	32.73	25.49	37.15	22.98	28.41
Baseline++	29.40	40.48	30.44	41.71	23.41	25.82
Meta-ProtoNet	**40.61**	**56.12**	**49.38**	**68.80**	**33.58**	**43.83**

Table 1: Average accuracy (%) comparison to state-of-the-arts with 95% confidence intervals on 5-way classification tasks under the cross-domain FSL setting. Best results are displayed in boldface.

Method	miniImageNet		CUB		SUN	
	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
Meta-Learner LSTM	24.99	29.79	36.23	44.39	30.99	44.86
MAML	45.69	60.90	48.87	63.99	57.75	71.45
Reptile	26.59	39.87	27.21	42.35	28.30	51.62
Matching Network	47.63	56.28	53.06	62.19	55.02	62.57
Prototypical Network	46.15	65.56	48.21	57.80	55.70	67.32
Relation Network	47.64	63.65	52.76	64.71	58.29	72.15
Baseline	23.84	32.09	25.14	35.35	27.44	34.54
Baseline++	30.15	41.19	32.48	42.43	35.56	44.42
Meta-ProtoNet	**47.87**	**66.05**	**53.30**	**65.37**	**58.79**	**73.90**

Table 2: Average accuracy (%) comparison to state-of-the-arts with 95% confidence intervals on 5-way classification tasks under the conventional FSL setting. Best results are displayed in boldface.

dation classes is used to evaluate the final reporting performance in the meta-test phase.

Evaluation Using the Conventional Setting. Table 1 shows the comparative results under the conventional FSL setting on three benchmark datasets. It is observed that Meta-ProtoNet outperforms the original Prototypical Network in all conventional FSL scenarios. For 1-shot and 5-shot on miniImageNet → miniImageNet, Meta-ProtoNet achieves about 1% higher performance than Prototypical Network. However, Meta-ProtoNet achieves 5% and 10% higher performance for 1-shot and 5-shot on CUB → CUB, and 3% and 6% higher performance on SUN → SUN. As the latter two scenarios are conducted on fine-grained classification datasets, we attribute the promising improvement to that the categories in these fine-grained datasets share more local concepts than those in coarse-grained datasets, and thus a more discriminative space can be rapidly learned with a few steps of adaptation. Moreover, Meta-ProtoNet achieves the best performance among all baselines in all conventional FSL scenarios, which shows that our approach can be considered as a better baseline option under the conventional FSL setting.

Evaluation Using the Cross-Domain Setting. We also conduct cross-domain FSL experiments and report the comparative results in Table 2. Compared to the results under the conventional setting, it can be observed that all approaches suffer from a larger discrepancy between the distributions of training and testing tasks, which results in a performance decline in all scenarios. However, Meta-ProtoNet still outperforms the original Prototypical Network in all cross-domain FSL scenarios, demonstrating that the bilevel optimization strategy for adaptation and the learning of transferable latent factors can be utilized to improve simple metric-based approaches. Also, Meta-ProtoNet achieves all the best results, indicating that our approach can be regarded as a promising baseline under the cross-domain setting.

4 Conclusion

In this paper, we propose Meta-ProtoNet to handle the challenge of heterogeneous task distributions in few-shot scenarios, aiming to learn a latent factor space in which metric-based classification of heterogeneous tasks can be better performed. Extensive experiments show that our proposed approach can be considered as a stronger baseline in both conventional and cross-domain few-shot settings.

References

Xiaobin Chang, Yongxin Yang, Tao Xiang, and Timothy M Hospedales. 2019. Disjoint label space transfer learning with common factorised space. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3288–3295.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. 2019. A closer look at few-shot classification. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 1126–1135.

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

Qi Li, Zhenan Sun, Ran He, and Tieniu Tan. 2017. Deep supervised discrete hashing. *arXiv preprint arXiv:1705.10999*.

Genevieve Patterson, Chen Xu, Hang Su, and James Hays. 2014. The sun attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision*, 108(1-2):59–81.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 4077–4087.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *NeurIPS*, pages 3630–3638.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. 2011. The caltech-ucsd birds-200-2011 dataset.

Reference and coreference in situated dialogue

Sharid Loáiciga[1] **Simon Dobnik**[2] **David Schlangen**[1]
[1]Computational Linguistics, Department of Linguistics, University of Potsdam, Germany
[2]CLASP, Department of Philosophy, Linguistics and Theory of Science,
University of Gothenburg, Sweden
{loaicigasanchez, david.schlangen}@uni-potsdam.de,
simon.dobnik@gu.se

Abstract

In recent years, a large number of corpora have been developed for vision and language tasks. We argue that there is still significant room for corpora that increase the complexity of both visual and linguistic domains and which capture different varieties of perceptual and conversational contexts. Working with two corpora approaching this goal, we present a linguistic perspective on some of the challenges in creating and extending resources combining language and vision while preserving continuity with the existing best practices in the area of coreference annotation.

1 Introduction

With the ease of combining representations from different modalities provided by neural networks, text and vision are coming together. There is a growing body of resources addressing a setting in which the visual context can be exploited to support a textual task, for example visual coreference resolution.

Several corpora have been developed in the domain of vision and language (V&L), for example corpora of image captions (Lin et al., 2014; Young et al., 2014; Krishna et al., 2017), images and paragraph descriptions (Krause et al., 2017), visual question answering (Antol et al., 2015), visual dialogue (Das et al., 2017) and embodied question answering (Das et al., 2018). Through these the V&L research has progressively moved from sentence descriptions to descriptions involving utterances and conversations, therefore adding complexity to their semantic representations. In parallel to the corpora, V&L systems have been developed but of course these are limited by the complexity of the task for which the dataset has been collected. The end goal of the current research is to move to a more complex linguistic setting involving multiparty dialogue and visual representations that go beyond individual images.

Situated reference resolution involves grounding linguistic expressions in perceptual representations (Harnad, 1990). Coreference resolution, traditionally a textual task, involves linking linguistic expressions referring to the same discourse entities (Stede, 2012). While challenging, the task is defined by the familiar nature of written texts: linear, planned and structured; defining thus the coreference mechanisms and devices found in them. In resources combining V&L, however, the textual part is often a dialogue or pairs of question-answers. As a result, the coreference devices differ considerably from those found in texts and are closer to actual conversations whereby people create reference to entities on the fly. This of course comes with its own challenges, but there are also some relations made easier since they can be grounded in the image.

As V&L come together, there is therefore an increased need for extending resources for the task of visual coreference resolution. This means engaging with the challenges along two axes:

- Dialogue: built by two speakers who each have their own mental state and cognitive process but who are communicating through referring expressions which are projected in the same conversation.
- Shared physical context: simultaneous access to an image or other perceptual context which enables non-linear references to it. Instead, the reference is guided by visual attention.

We present a linguistic perspective on these challenges by analysing a pilot annotation of two situated dialogue corpora: the *Cups* corpus (Dobnik et al., 2020) and the *Tell-me-more* corpus (Ilinykh et al., 2019), shown below in Figure 1 and example (1) respectively. Starting from the annotation scheme for several textual coreference datasets (Artstein and Poesio, 2006; Pradhan et al., 2007; Uryupina et al., 2019), this exercise proved useful to pinpoint in what ways the purely textual doc-

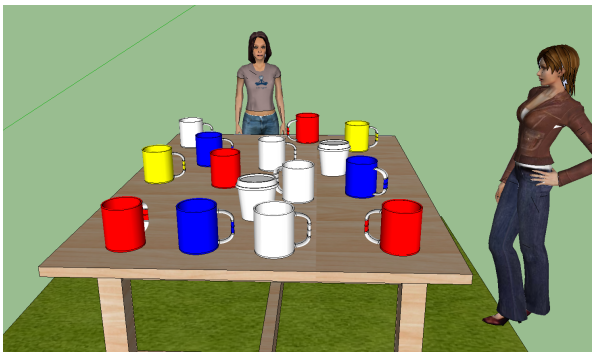
Proceedings of the Second Workshop on Advances in Language and Vision Research, pages 39–44

ument scenario is different from the domain of embodied interaction.

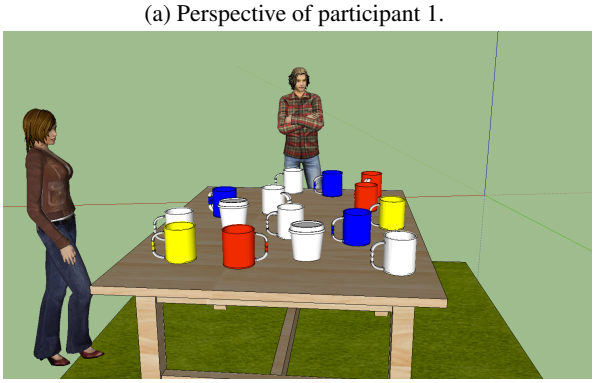
The first corpus contains a conversation between two participants over an almost identical visual scene involving a table and cups where participants have different locations (Figure 1). Some cups have been removed from each participant's view and they are instructed to discuss over a computer terminal in order to find the cups that each does not see. The *Tell-me-more* corpus consists of images accompanied with a small text of five complete sentences, collected by asking participants to describe the image to a friend, successively adding details. The genre of these texts is therefore mixed: in between standard text (as found in news text for example) and dialogue data which reflects the features found in conversations rather than written conventions.

These corpora are complementary as *Cups* gives us accurate visual ground truth information with free and unrestricted dialogue, while *Tell-me-more* offers a richer unrestricted image with short and task constrained (pseudo-)dialogues.

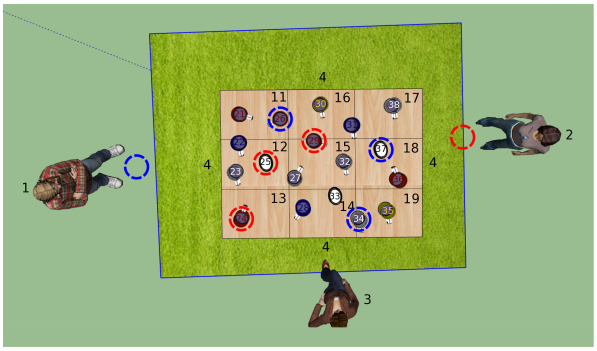
In this paper, we discuss a number of cases from these corpora that challenge both standard language grounding annotations as well as standard coreference annotation. This work points thus towards required future work in creating (co)reference annotation schemes that can handle situated dialogue.



(a) Perspective of participant 1.



(b) Perspective of participant 2.



(c) Top-down perspective of the Cups corpus scene with ground truth object IDs.

Figure 1: Participant 1 cannot see the cups circled in blue, whereas participant 2 cannot see the cups circled in red. Person 3 is visible to both participants as a reference point.

2 Related Work

Pointing to the inability of NLP tools to handle the textual part in situated dialogue, early works had described the need to ground the dialogue in the image in a manner informed by linguistics (Byron, 2003).

As content develops in a text, entities are introduced and re-mentioned, establishing discourse referents. The context is provided by the document and no extra-linguistic reference is needed for resolving the reference to an entity (Karttunen, 1969). In situated dialogue, on the other hand, the visual modality brings the extra-linguistic context as a source of referents. Here, resolving references to entities can be thus achieved by either looking at the picture or by reading the discourse. Recording both strategies separately is crucial if we want to understand and model them soundly, in keeping with theories of cognitive processing (cf. (Kelleher et al., 2005)). Extending the coreference annotation paradigm is thus the best bet although not a lot of work exists in this area.

Textual coreference Annotated data for the coreference resolution task has mainly focused on news texts and concrete nouns, excluding reference to events and other coreferential relations such as bridging, deixis, and ambiguous items well documented in the linguistic literature but deemed infrequent or too difficult to process (Poesio, 2016). In contrast, there is a growing body of literature interested in phenomena beyond the nominal case (Kolhatkar et al., 2018; Nedoluzhko and Lapshinova-Koltunski, 2016), resulting in new,

although still small in size, annotated corpora (Lapshinova-Koltunski et al., 2018; Zeldes, 2017; Uryupina et al., 2020).

Visual coreference Coreference work based on the popular VisDial dataset (Das et al., 2017) targets only a limited set of referential expressions, partly because it relies on automatic tools (Kottur et al., 2018; Yu et al., 2019), which are known to be problematic with this genre. With a focus in grounded human interaction, there are corpora whose textual part comprises question answer pairs (Antol et al., 2015; Goyal et al., 2017). Those, however, are short in nature, with few opportunities for re-mention of the different objects in the image and hence coreference. Last, corpora designed towards navigation and location (Stoia et al., 2008; Thomason et al., 2019) focusing on different kind of task and descriptions might be good candidates that could be explored and extended in a similar fashion as our corpora.

Referring expressions generation The goal in this area is to generate expressions over several turns of conversation in a natural and non-repetitive way, following principles of communicative discourse as for example in the recent PhotoBook dataset (Takmaz et al., 2020). Our work is complementary to such undertakings as it focuses on the interpretative rather than generative part.

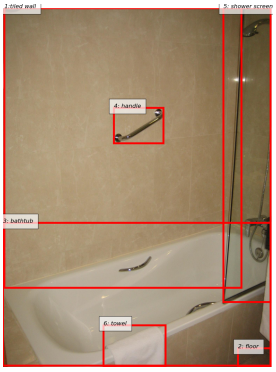
3 Understanding reference in situated dialogue

The notion of coreference chain–the sequence of mentions pointing to a same entity in a text–is central in coreference resolution. Built on top of the document as a unit, this notion relies on and in turn informs theories about accessibility hierarchy and salience of entities (Ariel, 1988, 2004; Grosz et al., 1995). In dialogue, however, references crisscross between the speakers and, one step further, in situated dialogue references crisscross between the speakers and the objects in the image. In this section we revise the annotation challenges in the annotation of anaphoric phenomena in data of this genre.

3.1 Grounding and referentiality

In spoken discourse people try their best to ground the references so they make sure they understand each other. To do so, they rely on the mechanisms of attention (Lavie et al., 2004). Although most concrete references can be grounded to the image easily, there are also some difficult cases. References can be found to portions of the image without a bounding box, such as *base of the tub* in example (1).

(1)



1. This is a picture of a bathtub.
2. The tub is white.
3. The wall and base of the tub are brown.
4. The door appears to be glass.
5. There is a handrail on the side wall.

In the previous example the difficulty arises because the object detector failed to recognise the target object. However, referring expressions are referential to a different degree, e.g., "Where are your blue ones?" – is the speaker referring to a particular subset of blue cups, all the blue cups in the scene, blue cups in general, or not referring to any particular set of objects? The distinction is sometimes not clear.

Last, as the image determines the scope of the referentiality, typical semantic properties are frequently used to refer back to the objects in the image: colour, shapes, sizes. These can be genuinely referential (a form of ellipsis) or used in an attributive manner. Compare for example *white* in the second sentence of (1), with (2) below.

(2) P1: closest to me, from left to right red, blue, white, red
P2: ok, on your side I only see red, blue, white

3.2 Speakers' cognitive state

Contrary to a Gricean-based analysis of spoken discourse, coherence-based theories of discourse do not traditionally take the cognitive state of the speaker as a necessary element to text interpretation (Bender and Lascarides, 2019). In situated dialogue, however, although the image can be treated as the ground truth of the situation, the speaker's cognitive state has to be considered to disambiguate their utterances, the hearer makes a model of their beliefs, desires and intentions associated with the utterance. This is exemplified in the following excerpt from *Cups* where both participants do not see one of the two red cups close by, but each a differ-

ent one. They mistakenly believe that there is only one missing red cup and this dis-alignment of their beliefs gradually leads to increasingly diverging cognitive states.

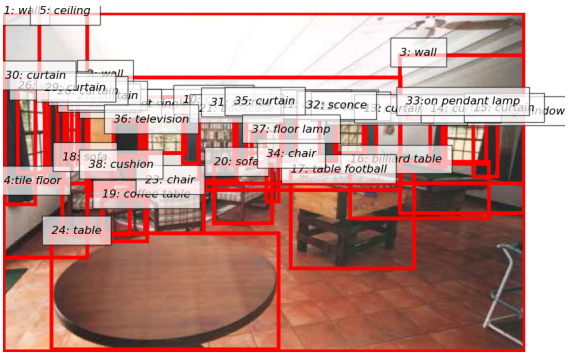
(3)
P2: there is an empty space on the table on the second row away from you
P2: between the red and white mug (from left to right)
P1: I have one thing there, a white funny top
P2: ok, i'll mark it.
DIALOGUE_STATE: B found O-25.
P1: and the red one is slightly close to you
P1: is that right?
P1: to my left from that red mug there is a yellow mug
P2: hm...
P2: can't see that and now i'm confused
DIALOGUE_STATE: B cannot see O-29.
P2: describe the second row away from you like you see it
P1: only one thing there, a white funny top
P2: aha, so it's closer to you than those i call "the second row"
P1: behind that, there is a yellow, red, white and blue
P1: from my left to right
P1: yes, that must be it!
P1: so what do you see in the "second row" from my perspective?
P2: i see a red, then space, then white and blue (same as katie's")
P2: no yellow
P2: is it on the edge of the table?
P2: on your left
P1: ok, yes!
DIALOGUE_STATE: inconsistent

3.3 Level of specification

We observe a common strategy of grouping things in order to refer to them collectively. This raises the question: What is the level of specification needed in a coreference annotation? One could think about this in linguistic terms, for instance mass nouns or sets; alternatively, in computer vision, there is the distinction between things and stuff (Caesar et al., 2018).

In (4) below, is the reference to the *curtains* a case of a set composed by individual instances, or is it a mass noun? Note the curtains is a type of stuff in Caesar et al.'s work.

(4)
1. I see a picture of an entertainment room. 2. there is a round table in the foreground and a fussball table in the middle of the room, as well as a pool table further back. 3. there is a sitting area with chairs facing a television set. 4. the room has several windows with green curtains. 5.the floors are made of a brown tile.



In (5) from Cups, on the other hand, the speakers refer to *rows* of objects even though these are not arranged in strict geometric lines. Hence, which objects are included is contextually defined and not always clear.

(5)
P2: ok, so your next row
P2: you said there 's a takeaway cup somewhere marooned all alone
P1: Okay. So we have that row I described with the now found red cup. Then a takeaway cup that is between that row and the next. It's very much in the middle of the two rows.

3.4 Information status

Different referring expressions have different properties and behaviour, an idea behind theories of salience and accessibility. They are based on the observation that some forms are used to introduce entities and some others to refer to them: some entities are discourse-new and some are discourse-old. In situated dialogue, the image provides an additional context and source of referents, but it does not follow that the status of subsequent mentions is *old*. In the example below, the fact that the discourse starts with *It* is licensed by the image and this source of reference should be accounted for differently in the annotation than a genuine

discourse-old case such as the *it* in sentence 2.

(6) 1. It s a well-lit kitchen with stainded [sic] wooden cupboards . 2. There's a microwave mounted over the stove, which has a red tea kettle on it. 3. The appliances are black and stainless steel in the kitchen. 4. The countertops look like they 're black granite. 5. The window has sunlight streaming in and it 's very brightly light.

4 Conclusions

V&L resources provide a unique opportunity to explore the notion of discourse entity in grounded context. Extending the coreference annotation to this domain is essential to understand the relationship between reference and coreference. The same mechanisms that humans adopt to solve coreference in the textual domain should underlay results in the V&L domain. Indeed, reference is underspecified in both modalities; any kind of information extraction from these domains will benefit from mechanisms that resolve this underspecification: capturing coreference is a door to capturing coherence. Furthermore, a rich annotation scheme leads to the development of corpora allowing the training of data driven systems for the V&L domain and social robotics.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual Question Answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Mira Ariel. 1988. Referring and accessibility. *Journal of Linguistics*, 24(1):65–87.

Mira Ariel. 2004. Accessibility marking: Discourse functions, discourse profiles, and processing cues. *Discourse Processes*, 37(2):91–116.

Ron Artstein and Massimo Poesio. 2006. Arrau annotation manual (trains dialogues).

Emily M. Bender and Alex Lascarides. 2019. Linguistic Fundamentals for Natural Language Processing II: 100 Essentials from Semantics and Pragmatics. *Synthesis Lectures on Human Language Technologies*, 12(3):1–268.

Donna K Byron. 2003. Understanding referring expressions in situated language some challenges for real-world agents. In *Proceedings of the First International Workshop on Language Understanding and Agents for Real World Interaction*, pages 39–47.

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. 2018. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. 2018. Embodied question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2054–2063.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. 2017. Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 326–335.

Simon Dobnik, John D. Kelleher, and Christine Howes. 2020. Local alignment of frame of reference assignment in English and Swedish dialogue. In *Spatial Cognition XII: Proceedings of the 12th International Conference, Spatial Cognition 2020, Riga, Latvia*, pages 251–267, Cham, Switzerland. Springer International Publishing.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1995. Centering: A framework for modelling the local coherence of discourse. *Computational Linguistics*, 2(21):203–225.

Stevan Harnad. 1990. The symbol grounding problem. *Physica D*, 42(1–3):335–346.

Nikolai Ilinykh, Sina Zarrieß, and David Schlangen. 2019. Tell me more: A dataset of visual scene description sequences. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 152–157, Tokyo, Japan. Association for Computational Linguistics.

Lauri Karttunen. 1969. Discourse referents. In *International Conference on Computational Linguistics COLING 1969: Preprint No. 70*, Sånga Säby, Sweden.

John D. Kelleher, Fintan J. Costello, and Josef van Genabith. 2005. Dynamically structuring updating and interrelating representations of visual and linguistic discourse. *Artificial Intelligence*, 167:62–102.

Varada Kolhatkar, Adam Roussel, Stefanie Dipper, and Heike Zinsmeister. 2018. Anaphora with non-nominal antecedents in computational linguistics: a survey. *Computational Linguistics*, 44(3):547–612.

Satwik Kottur, José M. F. Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. 2018. Visual coreference resolution in visual dialog using neural module networks. In *The European Conference on Computer Vision (ECCV)*.

Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. 2017. A hierarchical approach for generating descriptive image paragraphs. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3337–3345.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. 2017. Visual Genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73.

Ekaterina Lapshinova-Koltunski, Christian Hardmeier, and Pauline Krielke. 2018. ParCorFull: a parallel corpus annotated with full coreference. In *Proceedings of 11th Language Resources and Evaluation Conference*, pages 423–428, Miyazaki, Japan. European Language Resources Association (ELRA). To appear.

Nilli Lavie, Aleksandra Hirst, Jan W de Fockert, and Essi Viding. 2004. Load theory of selective attention and cognitive control. *Journal of Experimental Psychology: General*, 133(3):339–354.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.

Anna Nedoluzhko and Ekaterina Lapshinova-Koltunski. 2016. Abstract coreference in a multilingual perspective: a view on czech and german. In *Proceedings of the Workshop on Coreference Resolution Beyond OntoNotes*, CORBON 2016, pages 47–52, Ann Arbor, Michigan. Association for Computational Linguistics.

Massimo Poesio. 2016. Linguistic and cognitive evidence about anaphora. In Massimo Poesio, Roland Stuckardt, and Yannick Versley, editors, *Anaphora Resolution: Algorithms, Resources and Applications*, pages 23–54. Springer-Verlag, Berlin Heidelberg.

Sameer S. Pradhan, Lance Ramshaw, Ralph Weischedel, and Jessica MacBrideand Linnea Micciulla. 2007. Unrestricted coreference: Identifying entities and events in OntoNotes. In *International Conference on Semantic Computing (ICSC 2007)*, pages 446–453.

Manfred Stede. 2012. *Disourse Processing*. Morgan and Claypool Publishers, Toronto.

Laura Stoia, Darla Magdalene Shockley, Donna K. Byron, and Eric Fosler-Lussier. 2008. SCARE: a situated corpus with annotated referring expressions. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Ece Takmaz, Mario Giulianelli, Sandro Pezzelle, Arabella Sinclair, and Raquel Fernández. 2020. Refer, Reuse, Reduce: Generating Subsequent References in Visual and Conversational Contexts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4350–4368, Online. Association for Computational Linguistics.

Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. 2019. Vision-and-dialog navigation. In *Conference on Robot Learning (CoRL)*.

Olga Uryupina, Ron Artstein, Antonella Bristot, Federica Cavicchio, Francesca Delogu, Kepa Rodriguez, and Massimo Poesio. 2020. Annotating a broad range of anaphoric phenomena, in multiple genres: the ARRAU corpus. *Natural Language Engineering*, 26(1):95–128.

Olga Uryupina, Ron Artstein, Antonella Bristot, Federica Cavicchio, Francesca Delogu, Kepa J. Rodriguez, and Massimo Poesio. 2019. Annotating a broad range of anaphoric phenomena, in a variety of genres: the ARRAU corpus. *Natural Language Engineering*, 26(1):95–128.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Xintong Yu, Hongming Zhang, Yangqiu Song, Yan Song, and Changshui Zhang. 2019. What you see is what you get: Visual pronoun coreference resolution in dialogues. 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)*, pages 5123–5132, Hong Kong, China. Association for Computational Linguistics.

Amir Zeldes. 2017. The GUM corpus: creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Author Index