

# Paper-Studium und Umsetzung

Zweite Mini Challenge des Moduls Deep Learning an der FHNW

# Datensatz und Ziel

 Erstellung von Captions für Bilder

 Flickr8k


 Unterschiedliche Bilddimensionen, 3 Farbkanäle

 8091 Bilder (70% Train, 15% Val, 15% Test)

 5 Captions/Bild

# Hauptframeworks

 PyTorch

 PyTorch Lightning

 Weights & Biases

# Referenzpaper

cs.CV] 20 Apr 2015

## Show and Tell: A Neural Image Caption Generator

Oriol Vinyals  
Google

vinyals@google.com

Alexander Toshev  
Google

toshev@google.com

Samy Bengio  
Google

bengio@google.com

Dumitru Erhan  
Google

dumitru@google.com

### Abstract

*Automatically describing the content of an image is a fundamental problem in artificial intelligence that connects computer vision and natural language processing. In this paper, we present a generative model based on a deep recurrent architecture that combines recent advances in computer vision and machine translation and that can be used to generate natural sentences describing an image. The model is trained to maximize the likelihood of the target description sentence given the training image. Experiments on several datasets show the accuracy of the model and the fluency of the language it learns solely from image descriptions. Our model is often quite accurate, which we verify*

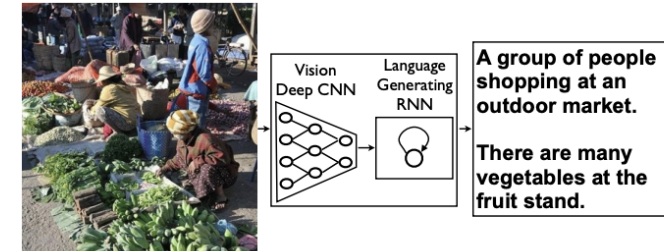
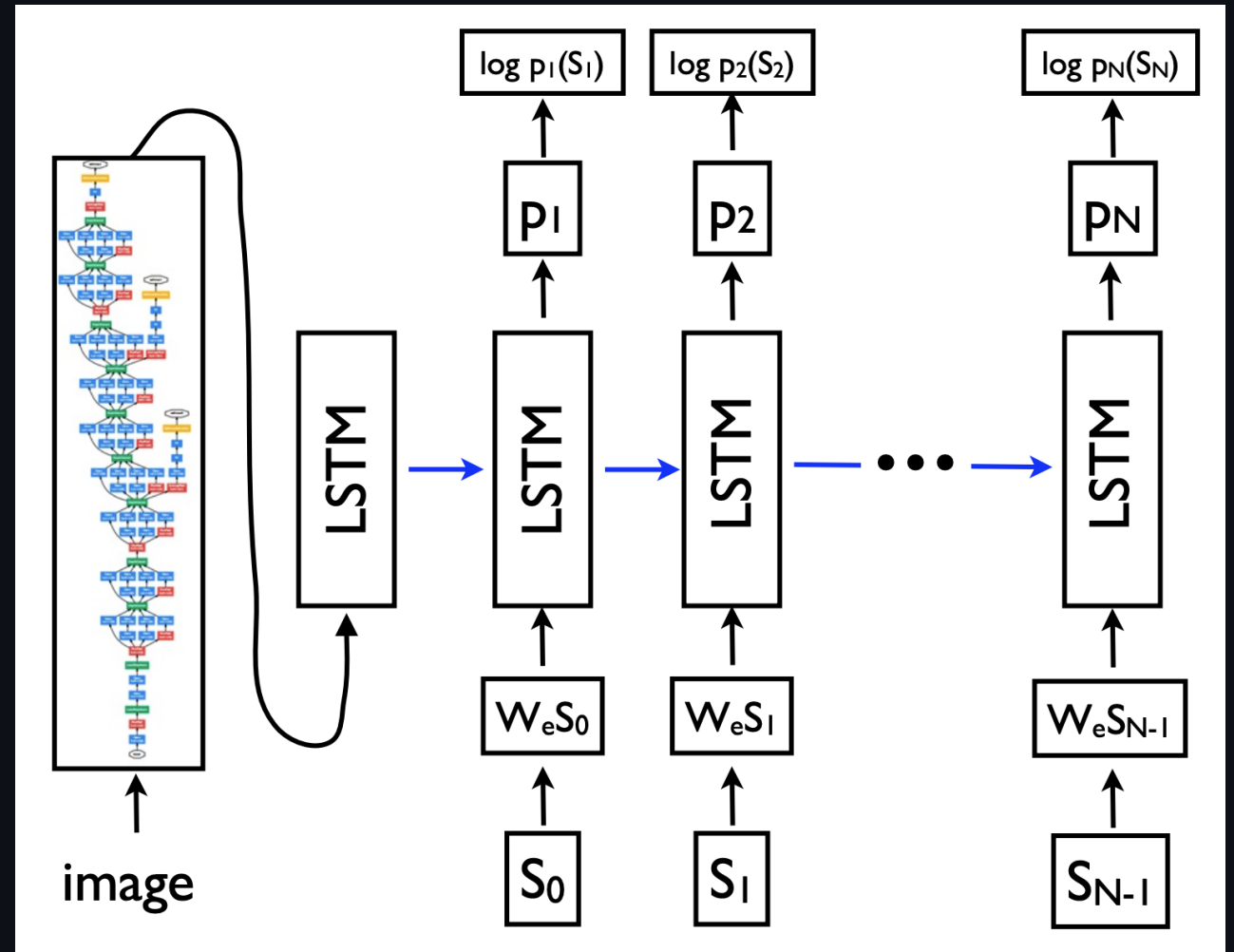


Figure 1. NIC, our model, is based end-to-end on a neural network consisting of a vision CNN followed by a language generating RNN. It generates complete sentences in natural language from an input image, as shown on the example above.

# Architektur

- Image Encoder (CNN)
- Caption Encoder (nn.Embedding)
- Captioning Decoder (LSTM)



# Verwendete Metriken: BLEU

## BLEU-1

- Überprüft Unigramme
- Korrekte Wörter vorhanden?

## BLEU-2,3 und 4

- Überprüft Bigramme, Trigramme, Quadrigramme
- Korrekte Wortreihenfolge?

# Image Preprocessing

```
transform = transforms.Compose(  
    [  
        transforms.ToPILImage(),  
        transforms.RandomHorizontalFlip(),  
        transforms.RandomRotation(10),  
        transforms.ColorJitter(brightness=0.2, contrast=0.2,  
                                saturation=0.2, hue=0.1),  
        transforms.Resize((224, 224)),  
        transforms.RandomResizedCrop(224, scale=(0.8, 1.0)),  
        transforms.ToTensor(),  
    ]  
)
```

# Text Preprocessing

Example Caption:

```
A child in a pink dress is climbing up a set of stairs in an entry way .
```

Tokenized caption (with nltk word tokenizer):

```
[<start>', 'A', 'child', 'in', 'a', 'pink', 'dress', 'is', 'climbing',  
'up', 'a', 'set', 'of', 'stairs', 'in', 'an', 'entry', 'way', '.', '<end>',  
'<pad>', '<pad>', ...]
```

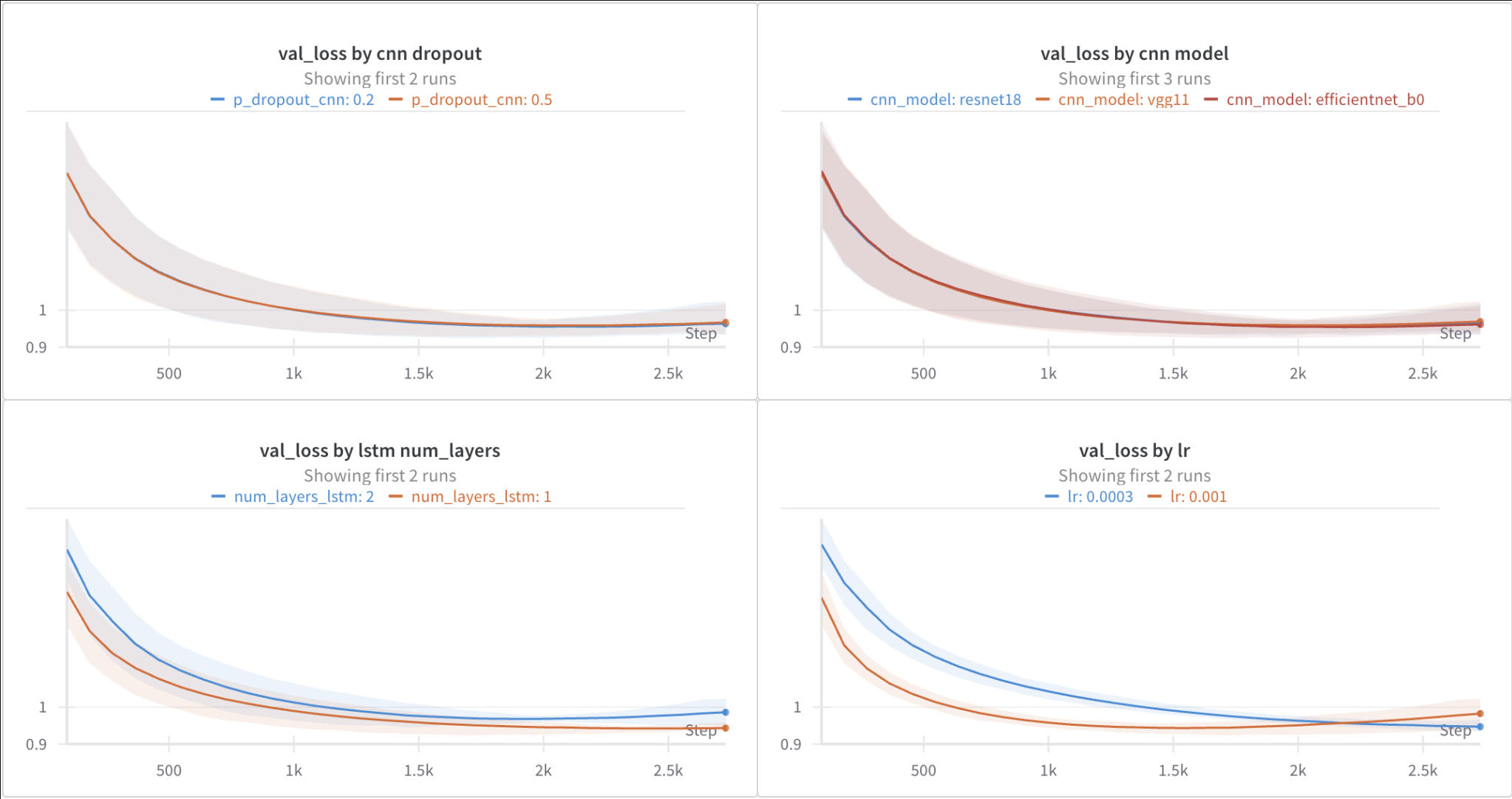
Tokenized caption (index in dictionary):

```
[9630, 68, 2580, 4910, 1240, 6514, 3457, 4995, 2675, 9105, 1240,  
7526, 6030, 8185, 4910, 1411, 3665, 9334, 13, 9631, 9632, 9632, ...]
```



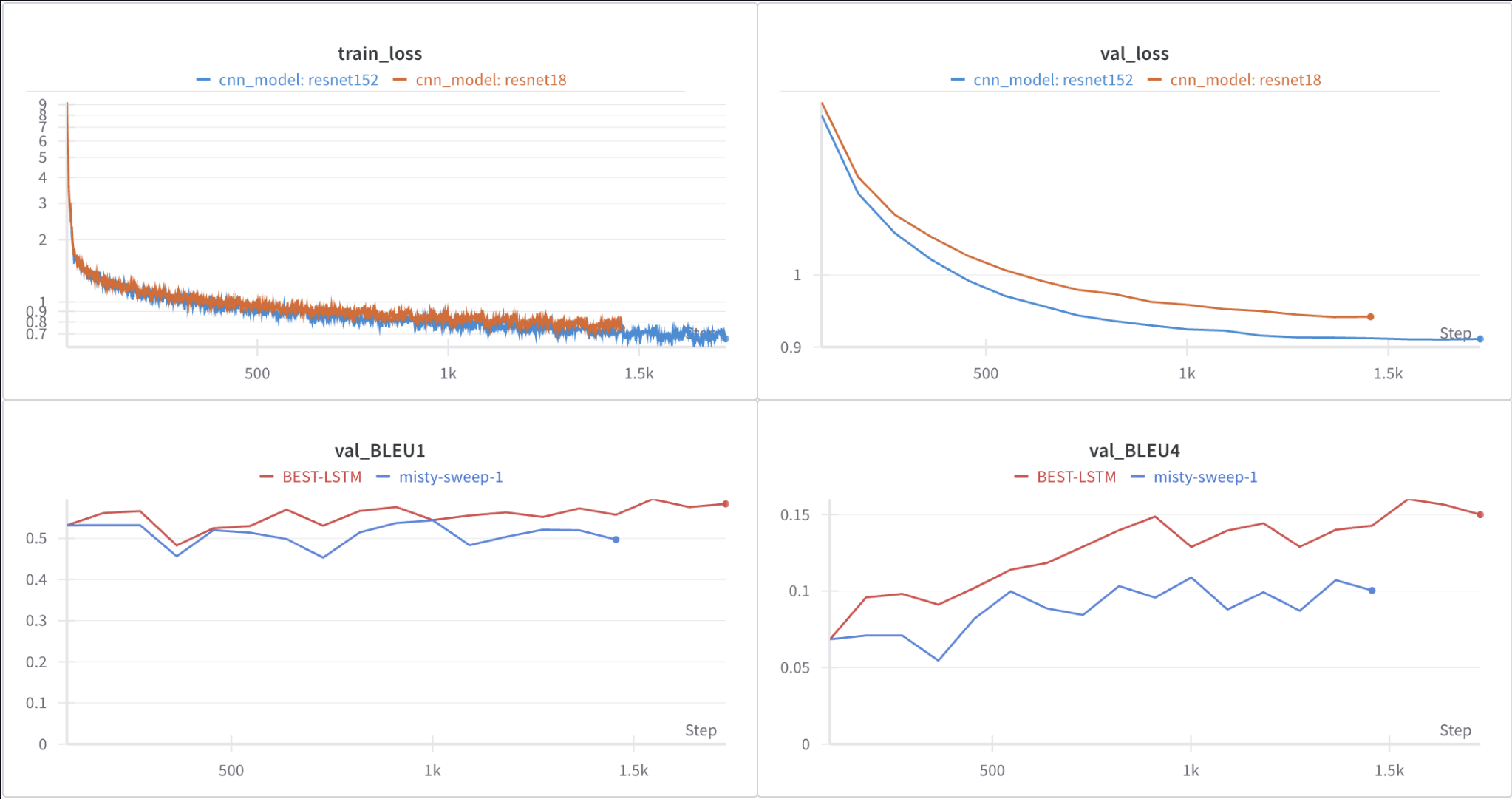
# Sweep 1 (Verhalten der Hyperparameter)

```
sweep_config = {
    "method": "grid",
    "name": "ShowAndTell",
    "parameters": {
        "optimizer": {"values": ["Adam"]},
        "lr": {"values": [0.001, 0.0003]},
        "weight_decay": {"values": [0.00001]},
        "cnn_model": {"values": ["efficientnet_b0", "vgg11", "resnet18"]},
        "embed_size": {"values": [512]},
        "p_dropout_cnn": {"values": [0.5, 0.2]},
        "hidden_size_lstm": {"values": [512]},
        "num_layers_lstm": {"values": [1, 2]},
        "n_epochs": {"values": [30]},
    },
}
```



## Sweep 2 (Vergleich der Modellgrösse)

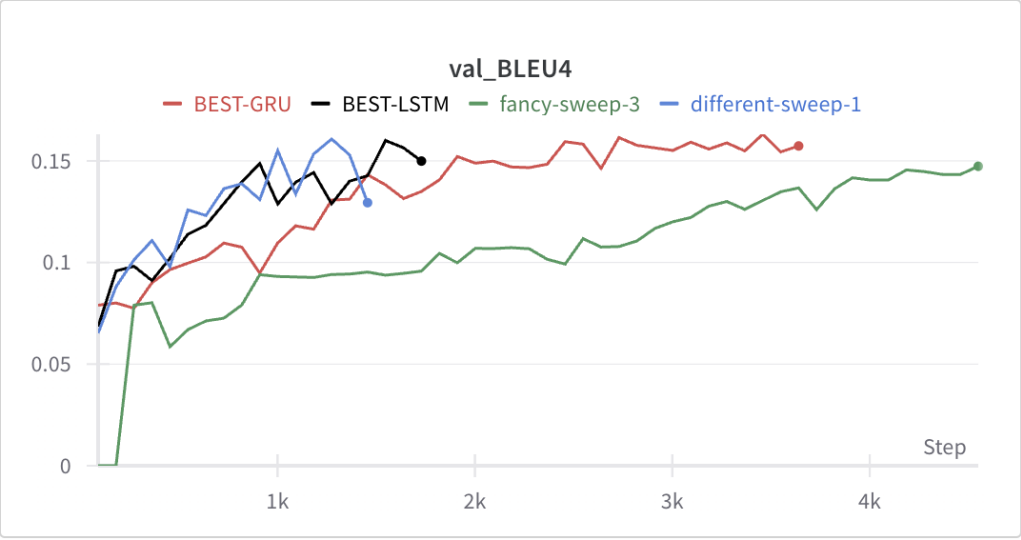
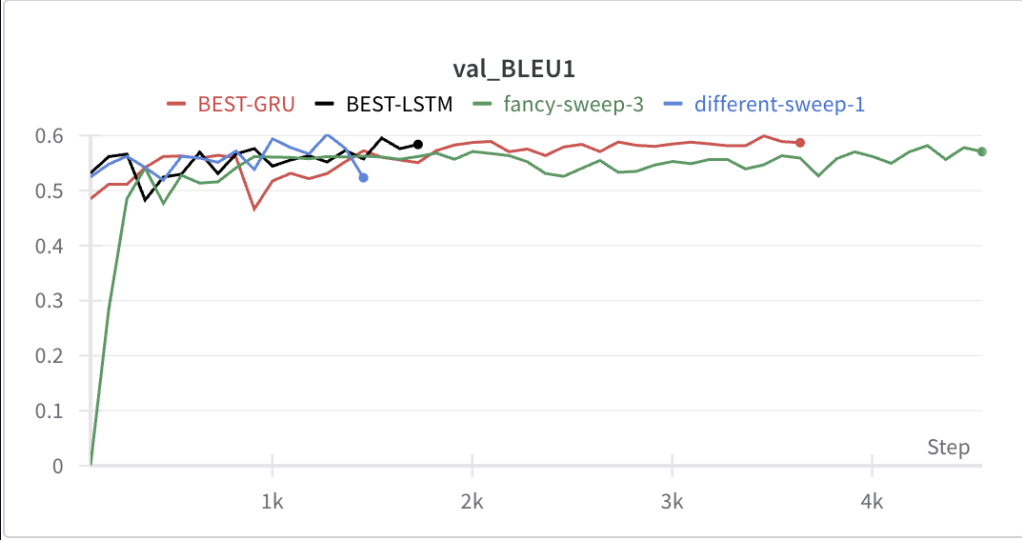
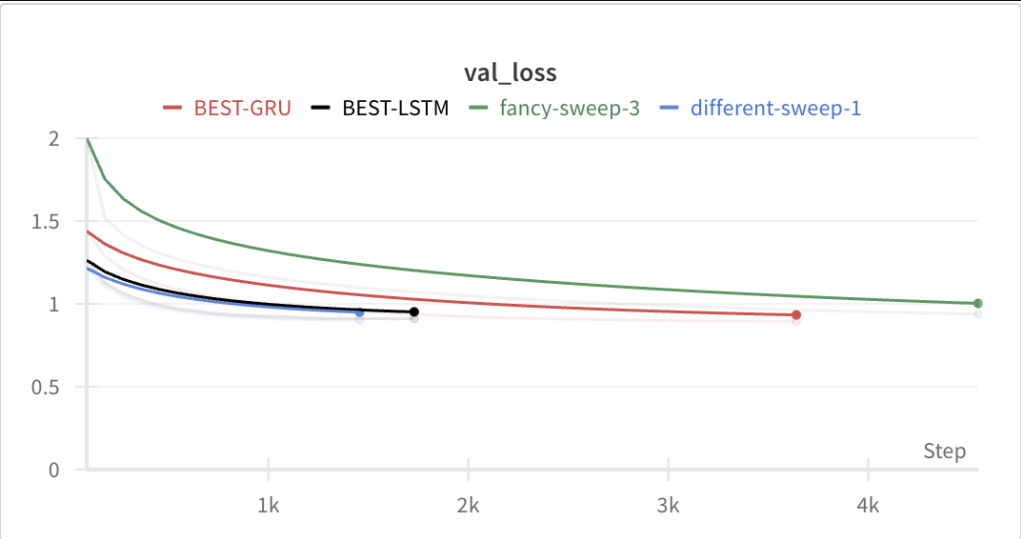
```
sweep_config = {  
    "method": "grid",  
    "name": "ShowAndTell2", # changed  
    "parameters": {  
        "optimizer": {"values": ["Adam"]},  
        "lr": {"values": [0.001]}, # changed  
        "weight_decay": {"values": [0.00001]},  
        "cnn_model": {"values": ["resnet18", "resnet152"]}, # changed  
        "embed_size": {"values": [512]},  
        "p_dropout_cnn": {"values": [0.5]}, # changed  
        "hidden_size_lstm": {"values": [512]},  
        "num_layers_lstm": {"values": [1]}, # changed  
        "n_epochs": {"values": [30]},  
    },  
}
```



epoch	caption	prediction
0	A black and white dog is running in a grassy garden surrounded by a white fence .	scanner bikinis competeition zombie background cheers twos videocameras rustic bullfight rollerblade rainbow plucking consoling smal Riwal ravine moving Greyhound Fawkes plaza gold woody interviewed hoops fan curtsey snowfall Rover lounging Stacks apple kangaroo VW piles opened mouthing featuring pinwheel
9	A black and white dog is running in a grassy garden surrounded by a white fence .	A black and white dog is running through the grass .
19	A black and white dog is running in a grassy garden surrounded by a white fence .	A dog is running through a field .

## Sweep 3 (GRU anstatt LSTM)

```
sweep_config = {  
    "method": "grid",  
    "name": "ShowAndTellGRU", # changed  
    "parameters": {  
        "optimizer": {"values": ["Adam"]},  
        "lr": {"values": [0.001, 0.0003, 0.0001]}, # changed  
        "weight_decay": {"values": [0.00001]},  
        "cnn_model": {"values": ["resnet152"]},  
        "embed_size": {"values": [512]},  
        "p_dropout_cnn": {"values": [0.5]},  
        "hidden_size_gru": {"values": [512]},  
        "num_layers_gru": {"values": [1]},  
        "n_epochs": {"values": [50]},  
    },  
}
```



## Metriken (Bestes LSTM)

	Datensatz	Unser Modell	Referenzmodell
BLEU-1	Flickr8k	0.5923	0.63
BLEU-2	Flickr8k	0.3716	:(
BLEU-3	Flickr8k	0.2390	:(
BLEU-4	Flickr8k	0.1555	:(
BLEU-4	MSCOCO (123'287 images)	-	0.277



## Gelungenes Beispiel



Caption: A brown dog on a leash runs through the white water .

Caption: A soaked dog is playing in the water .

Caption: A tan dog on a leash running in shallow ocean water .

Caption: A wet dog on a leash is running through some water .

Caption: Brown dog running through shallow water .

Prediction: A brown dog is running through the water .

---

## Ungelungenes Beispiel



Caption: A person eats takeout while watching a small television .

Caption: A person sits on the floor and eats in front of a television .

Caption: A television with a picture of a girl on it .

Caption: A young man sits on the floor by the television with a fast food meal in front of him .

Caption: Someone is laying in front of the TV eating food .

Prediction: A man is sitting on a couch .

---

## Mögliche Verbesserungen

- Tiefere Lernrate, dafür mehr Epochen
- Mehr Daten
- Stärkere Image Encoder (Panoptische Segmentierung?)
- Beam Search

## Lessons Learned

! Image Encoder wichtig!

⊘ Captioning Decoder eher unwichtig..

👍 Teacher Forcing für Textgeneration

! BLEU-1 Metrik ergänzen!