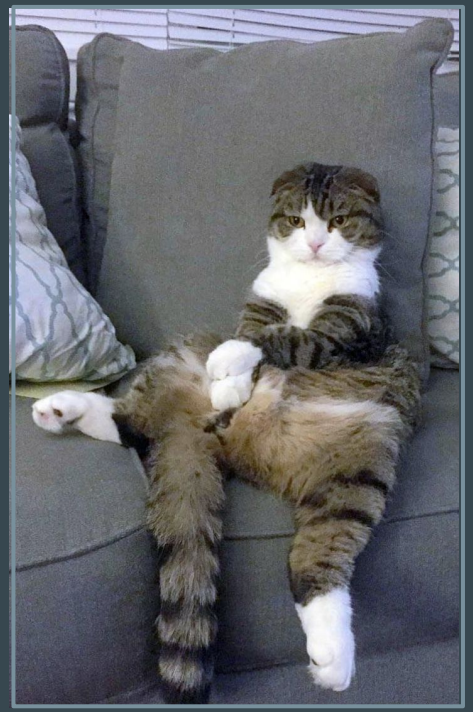
Attention Based Image Captioning in a Merge Architecture

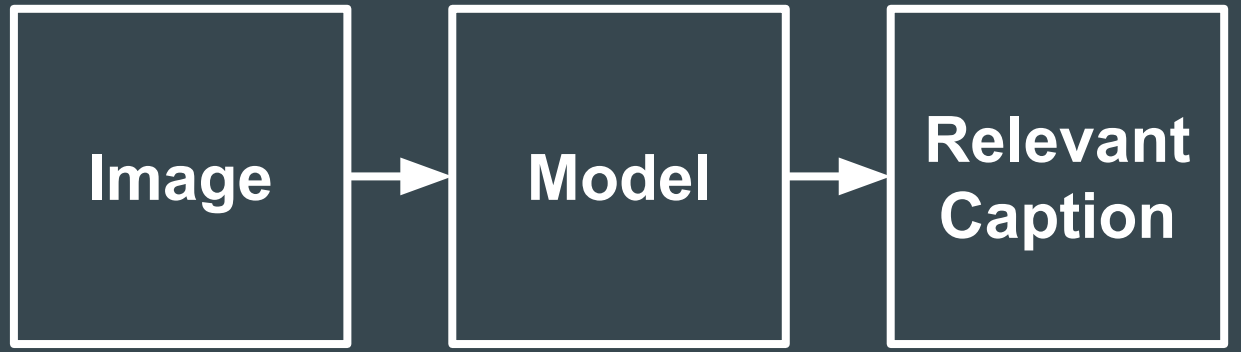
•••

W266 Final Project

Catherine Cao

What is Image Captioning?

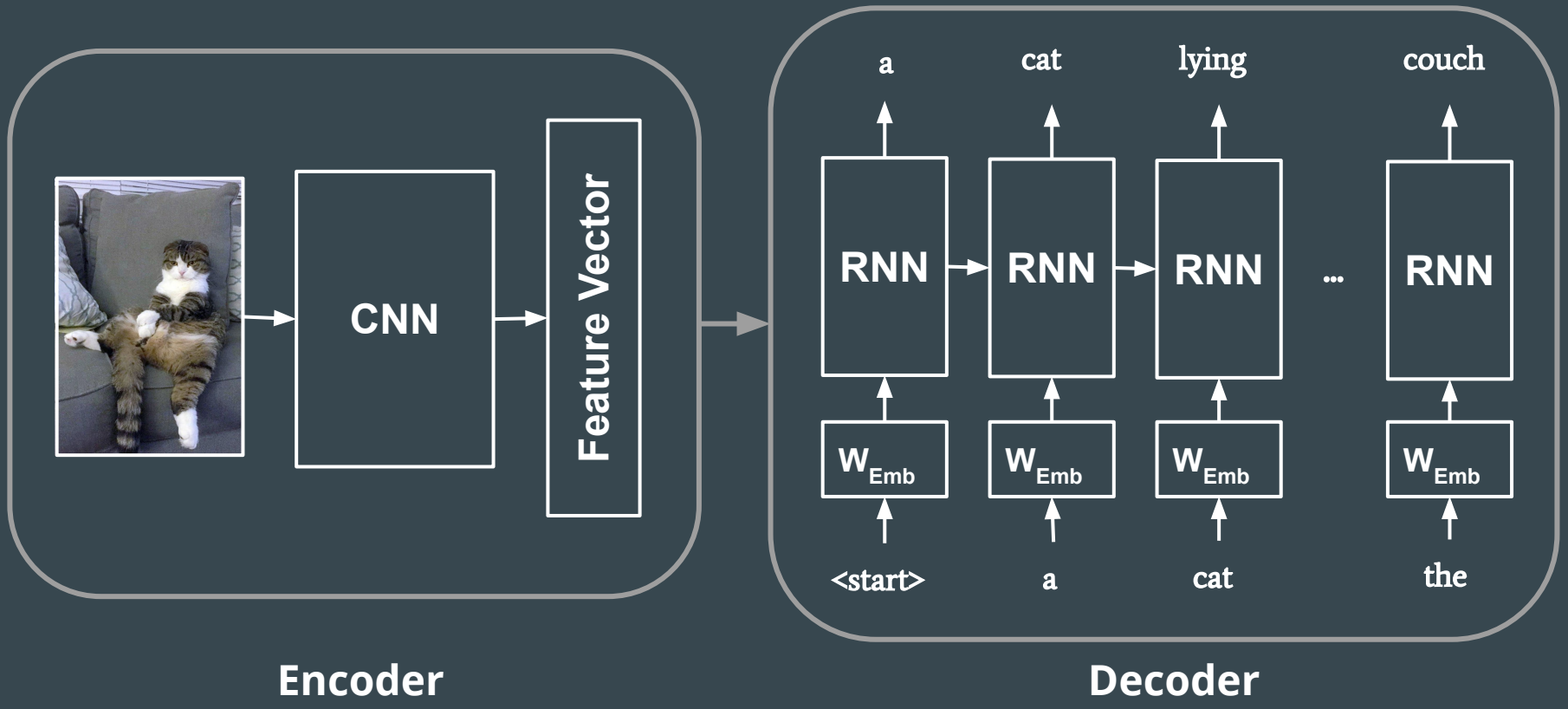




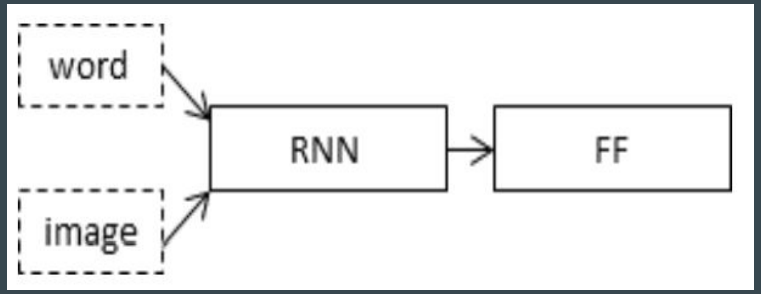
“Image caption generation is the task of generating a natural language description of the content of an image.“

(Bernardi et al., 2016)

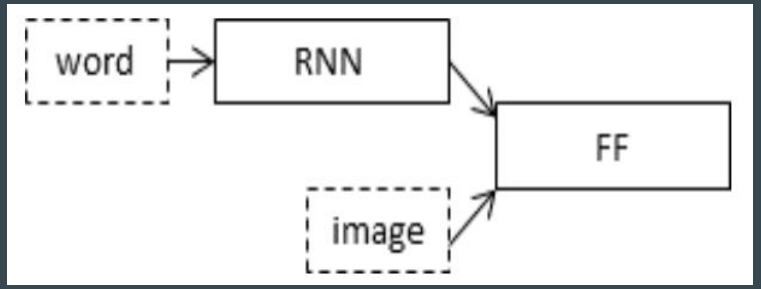
The General Approach



Inject vs. Merge Architecture

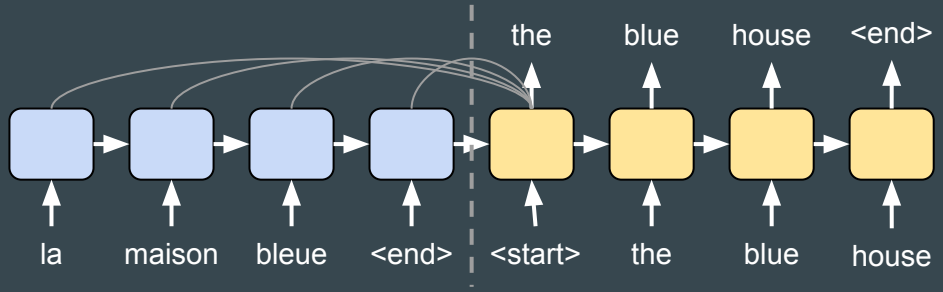


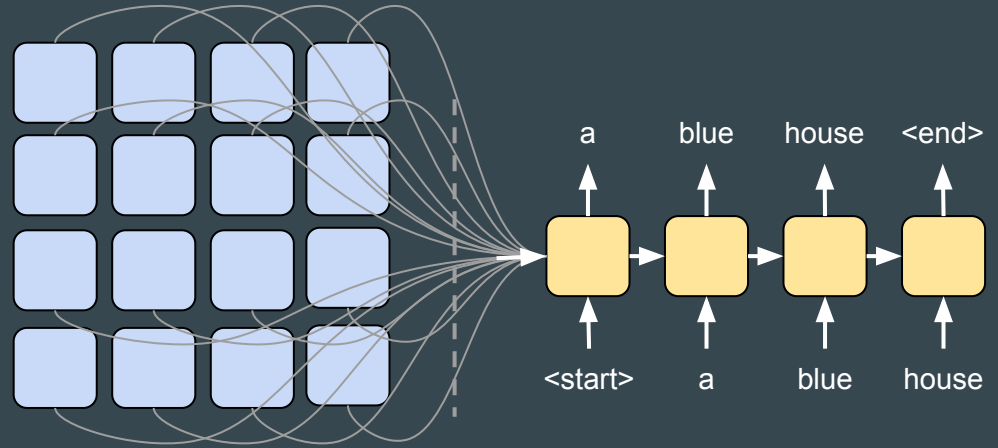
Inject Architecture



Merge Architecture

Attention Mechanism in Image Captioning

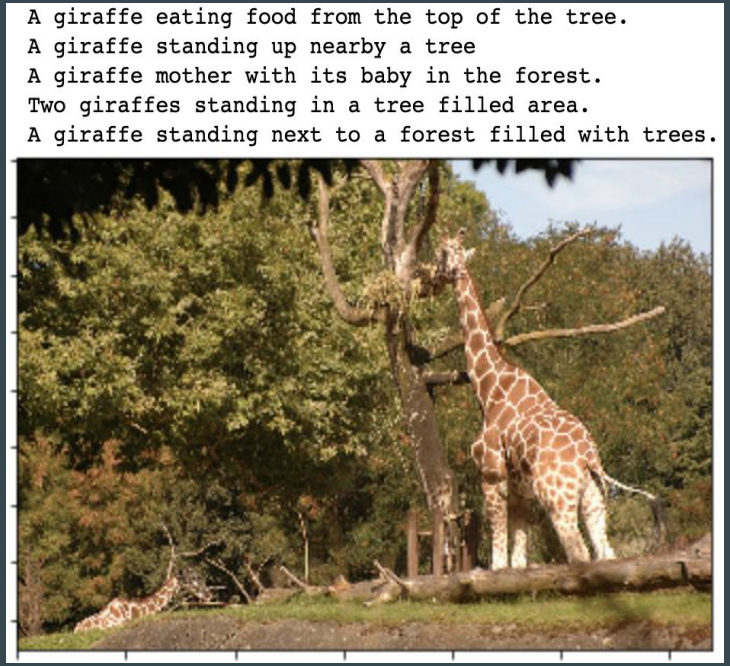




Data

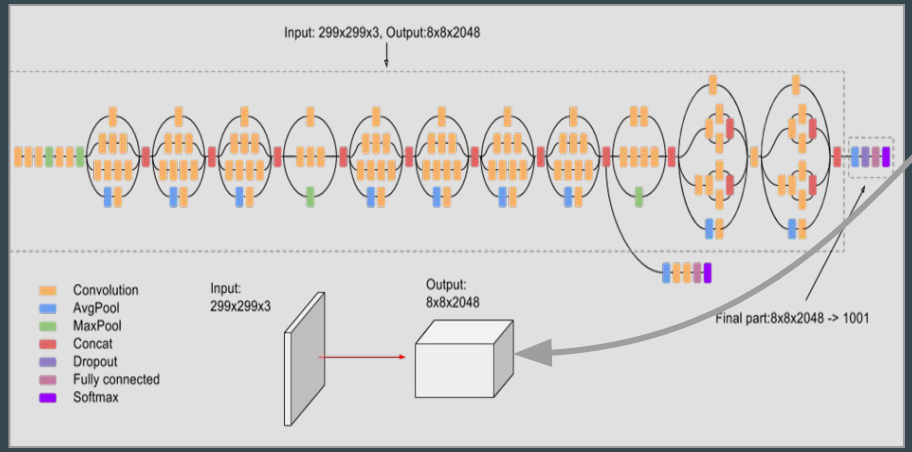
COCO 2014 Dataset Training Set:
82,723 images, 5 captions per image
Total of 414,114 unique image/captions

- Training: 30,000 images with unique captions
- Validation: 5,000 images with unique captions
- Test: 5,000 images with unique captions



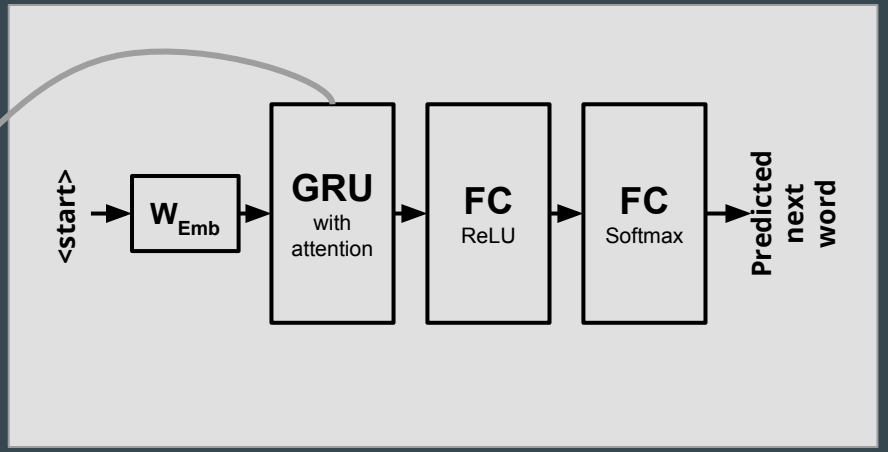
Model

Encoder



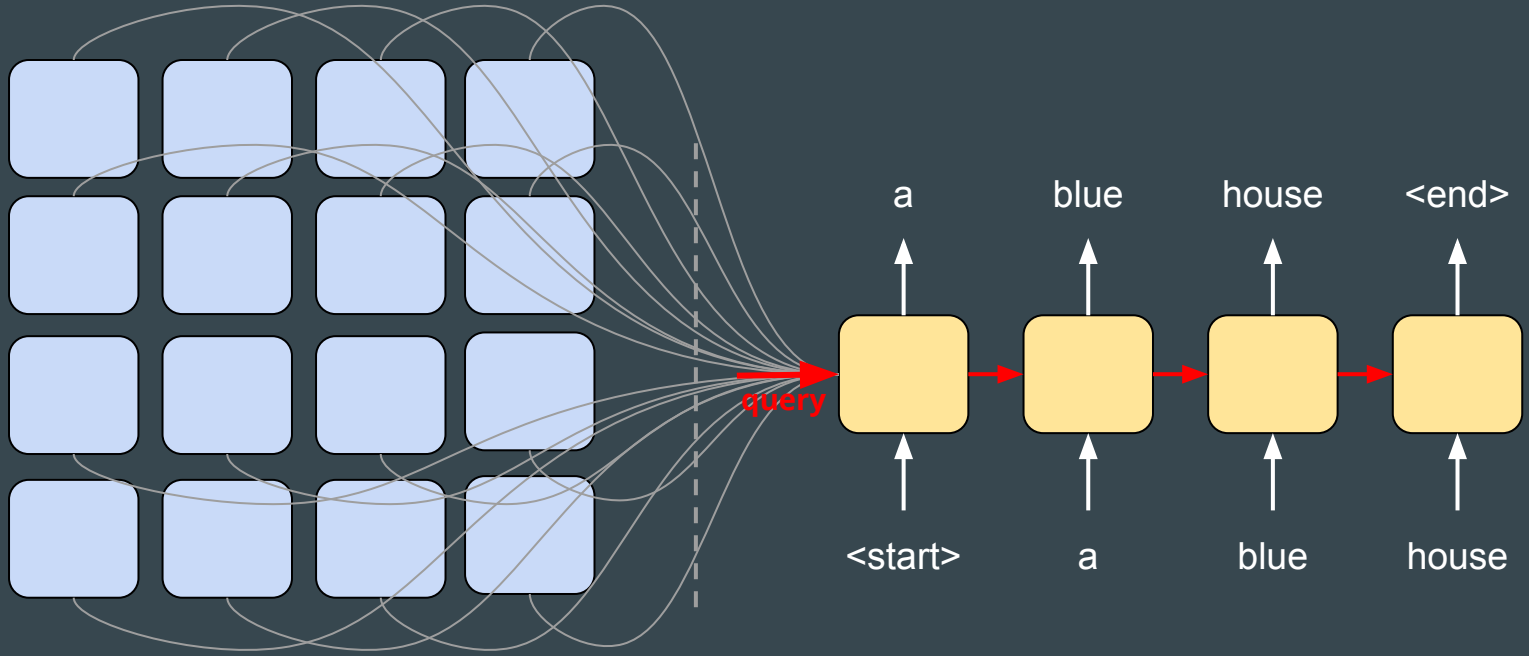
Last layer of InceptionV3 ImageNet model as input
(8, 8, 2048) -> fully connected layer to (8, 8, 200)
Attention is given to the 8x8 grid

Decoder



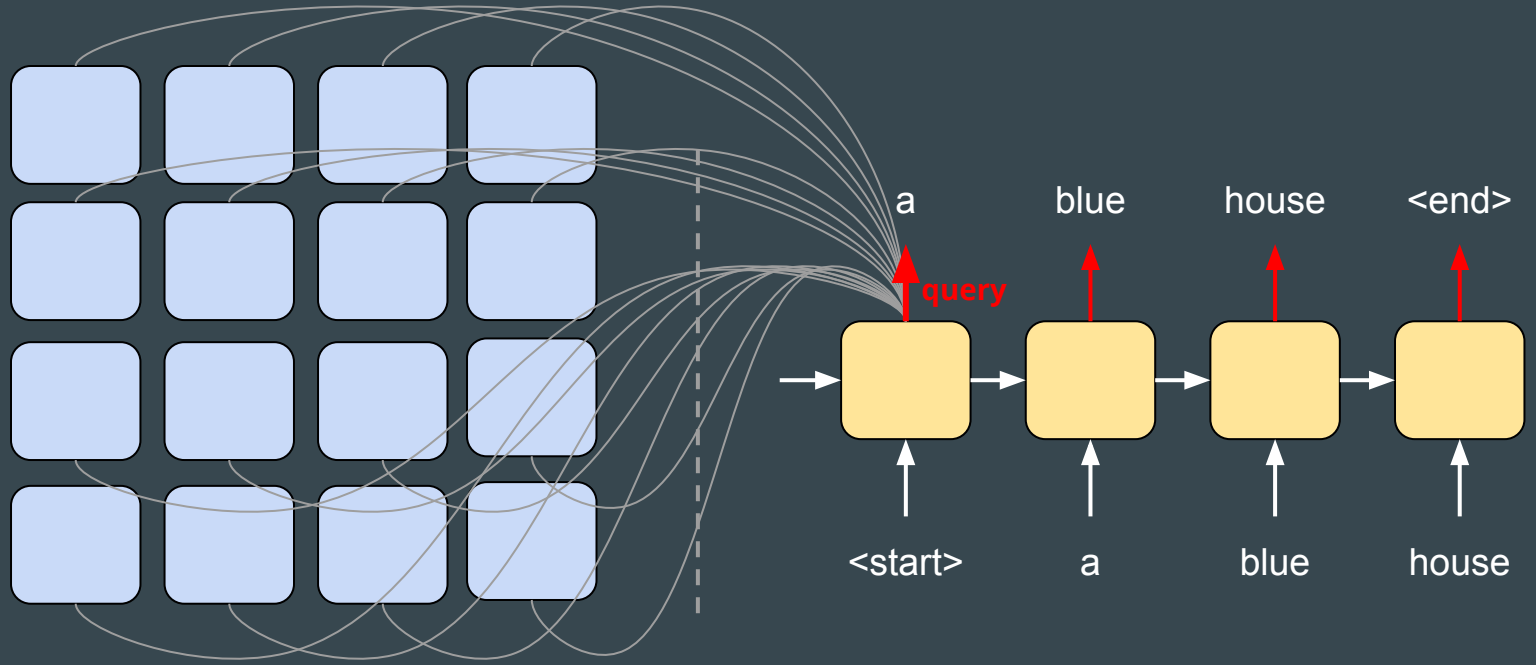
Word embeddings with 200 dimensions
Single GRU layer with 512 units
Two fully connected layer with softmax at the end

Attention in Inject Architecture



Hidden state of GRU as query to attend over the 8x8 grid of image matrix
Resulting context vector is concatenated to original hidden state

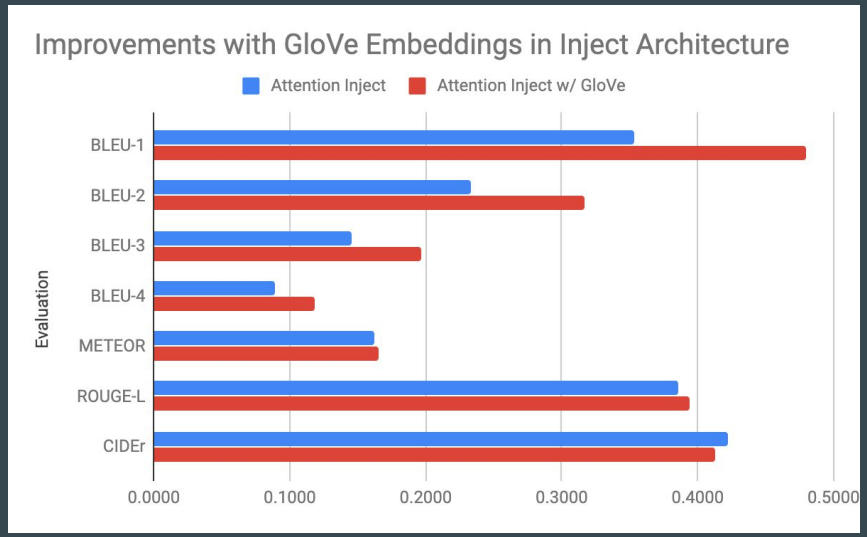
Attention in Merge Architecture

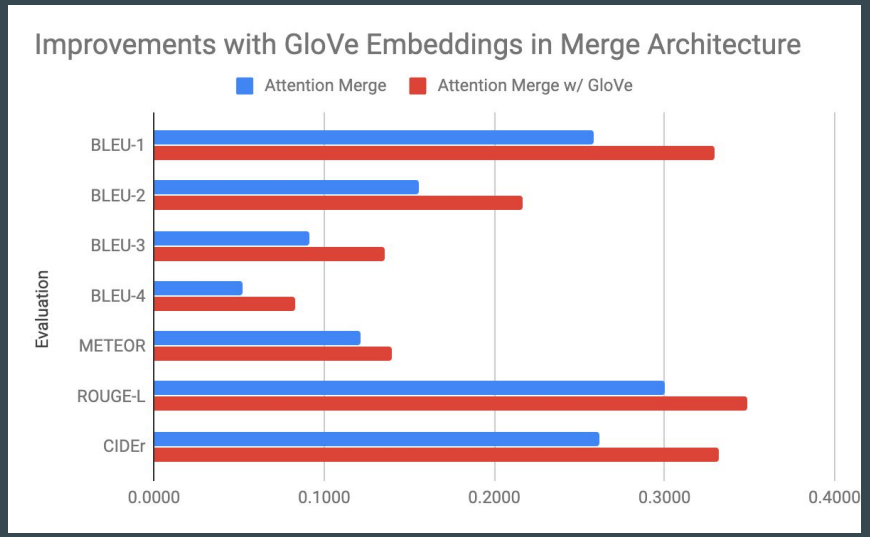


Output from GRU as query to attend over the 8x8 grid of image matrix
Resulting context vector is concatenated to output vector

Pre-trained Word Embeddings

- GloVe embeddings with 200 dimensions were used without further fine tuning
- Pre-trained embeddings improved results by 27% on average



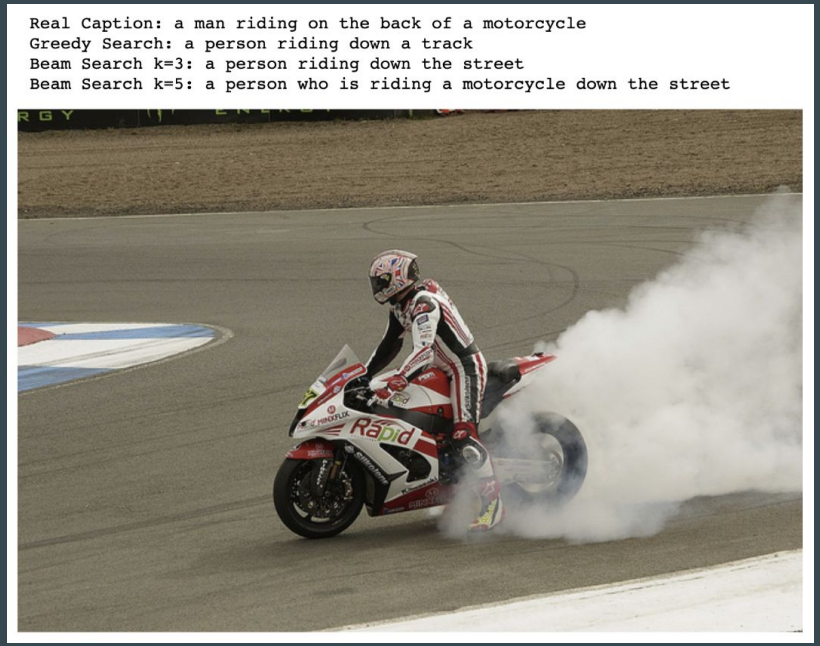


Beam Search

- A length-weighted beam search is implemented at inference to predict captions
- Probability of the entire sequence is calculated as:

$$P_{sequence} = \frac{\sum_{i=0}^{n} log(P_i)}{n}$$

- Beam search k=3 improves results by 8%
- Higher value of k has diminishing returns



Results

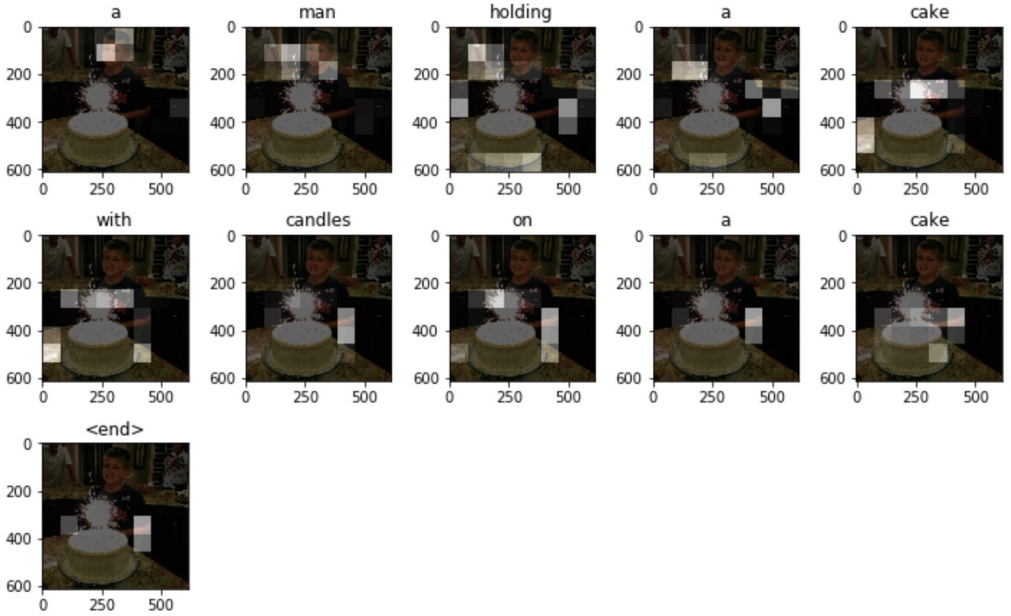
Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr
Attention Inject Baseline	0.3531	0.2327	0.1459	0.0889	0.1621	0.3852	0.4225
Attention Inject w/ GloVe[2]	0.4797	0.3170	0.1969	0.1185	0.1648	**0.3940**	0.4124
Attention Inject w/ GloVe Beam-3	**0.4894**	**0.3211**	**0.2053**	**0.1272**	**0.1664**	0.3860	**0.4232**
Attention Inject w/ GloVe Beam-5	0.4737	0.3069	0.1955	0.1196	0.1649	0.3791	0.4112
Attention Merge Baseline	0.2582	0.1560	0.0914	0.0518	0.1216	0.3001	0.2614
Attention Merge w/ GloVe[3]	0.3295	0.2164	0.1356	0.0827	0.1395	0.3486	0.3319
Attention Merge w/ GloVe Beam-3	0.3937	0.2595	0.1679	0.1043	0.1432	0.3523	0.3549
Attention Merge w/ GloVe Beam-5	0.3787	0.2455	0.1583	0.0986	0.1393	0.3424	0.3392

- Inject architecture outperforms merge architecture in the attention framework
- Pre-trained embeddings greatly improves results and reduces training time
- Beam search with k=3 has the most benefit, k=5 has diminishing returns

[2,3] Also includes adding a ReLU activation in the fully connected layer

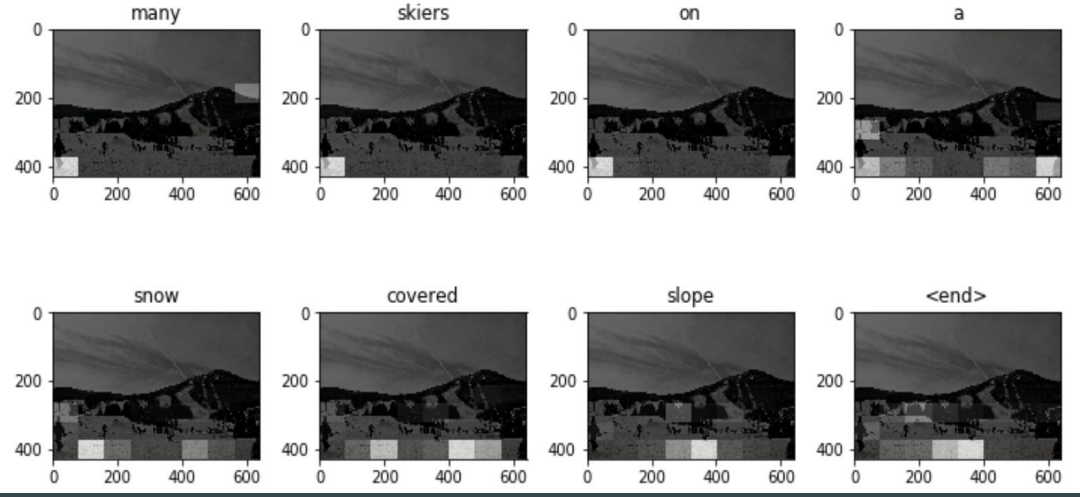
Results

True Caption: <start> a boy smiling over a birthday cake on a kitchen counter <end>
Predicted Caption: a man holding a cake with candles on a cake <end>



Results

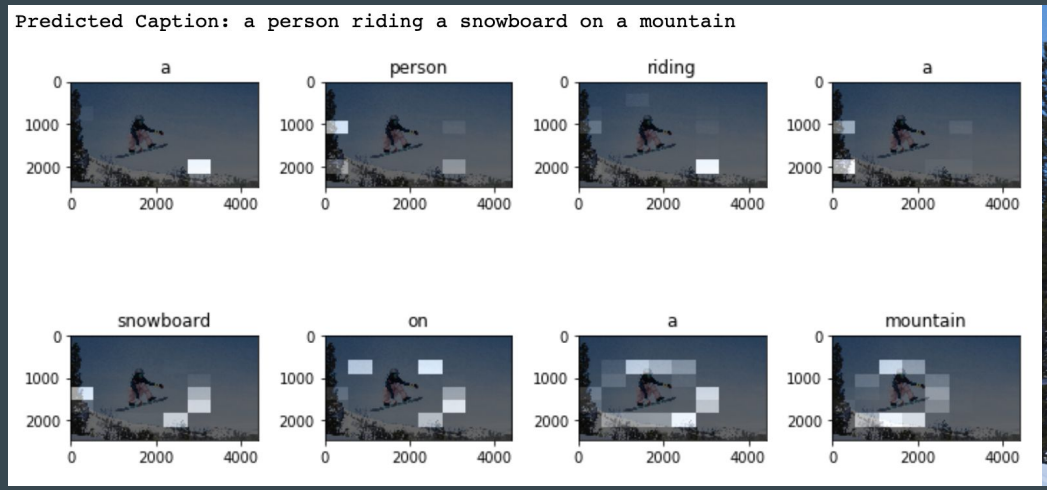
True Caption: <start> skiers walking around at a ski area <end>
Predicted Caption: many skiers on a snow covered slope <end>



Discussion

- Why does merge architecture not work well in attention framework?
- Captions with persistent phrases repeating itself
- Leverage pre-trained RNN in merge architecture
- How are the two architectures related to how human process image/text?
- Future works
  - Train on more data
  - Embeddings with context, e.g. BERT
  - Hyper-parameter tuning
  - Other evaluation metrics that measure diversity of words

Questions?


Predicted Caption: a person riding a snowboard on a mountain
a
person
riding
a
snowboard
on
a
mountain
0
1000
2000
0
2000
4000