Knowing When to Look: Adaptive Attention

Karim Ghonim
Hossam Arafat

Sapienza Università di Roma

Deep Learning for Computer Vision

Introduction

The purpose of this work is to propose a novel adaptive attention model with a visual sentinel

- Inspired by “Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning”

- Main points in this work:
  - network architecture
  - dataset
  - preprocessing
  - implementation
  - results

Motivation

- most methods force visual attention to be active for every generated word
- not all words in the caption have corresponding visual signals
- e.g., “sign” after “on top of a red stop”

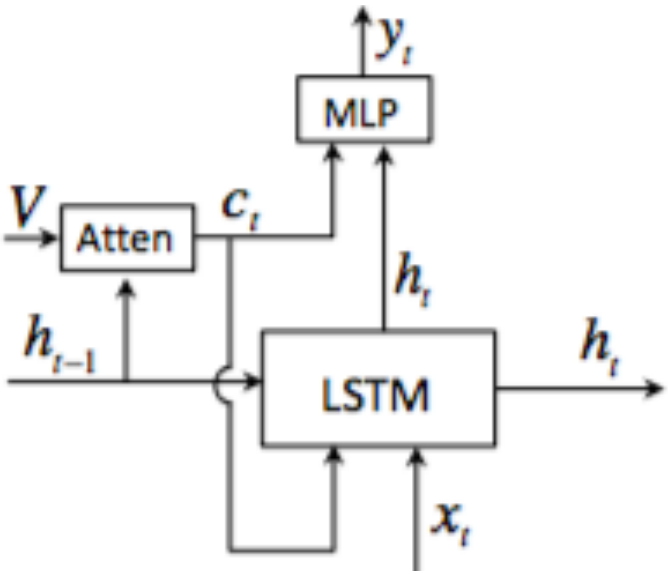
Motivation

- adaptive encoder-decoder framework that automatically decides when to look at the image and when to rely on the language model to generate the next word

- when relying on visual signals, the model also decides where – which image region – it should attend to

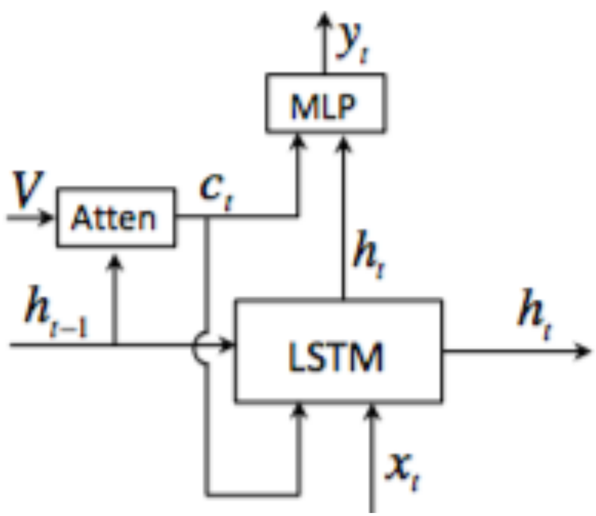
Network architecture

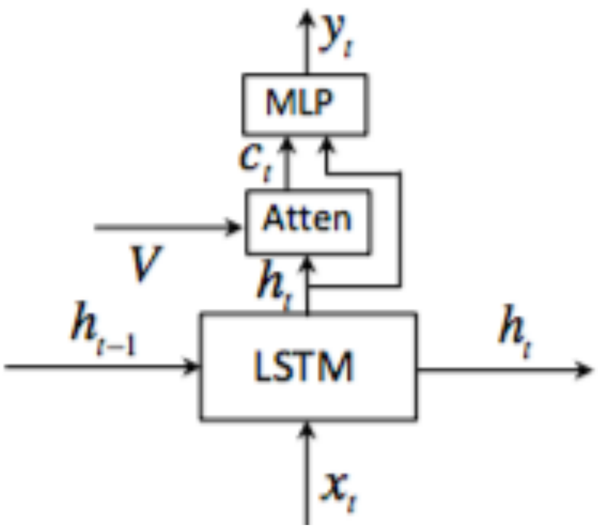
- Encoder-Decoder
  - CNN used as encoder
  - LSTM used as decoder
  - context vector in vanilla framework dependent only on encoder
  - context vector in attention-based framework dependent on both encoder and decoder



Network architecture

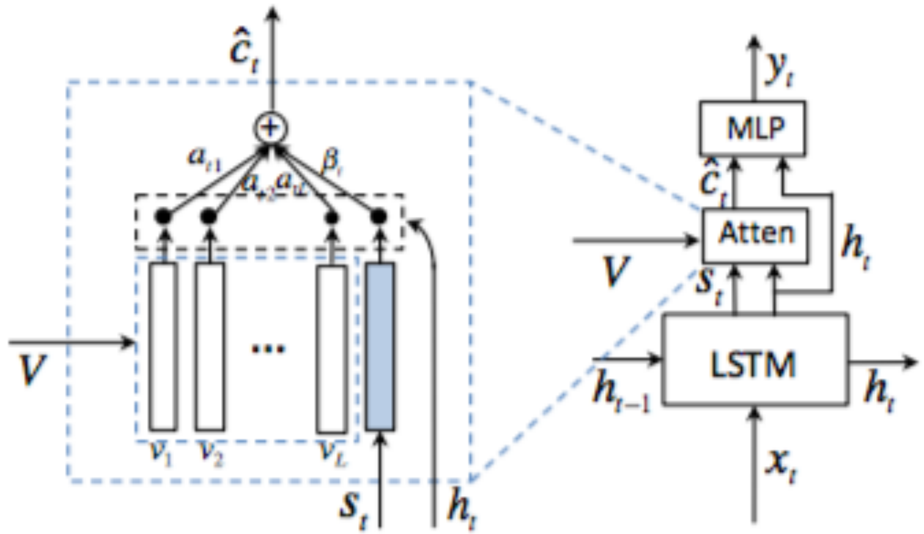
- Spatial Attention
  - context vector is generated using the current hidden state and the spatial image features
  - context vector considered as the residual visual information of current hidden state
  - complements the informativeness of the current hidden state for next word prediction





Network architecture

- Adaptive Attention
  - improves on the spatial attention model
  - relies on a new concept - “visual sentinel”
  - introduces novel way in generating context vector



Network architecture

- Visual Sentinel
  - latent representation of what the decoder already knows
  - the model can fall back on it when it chooses not to attend to the image
  - the gate that decides between attending to the image or the visual sentinel is called “sentinel gate”

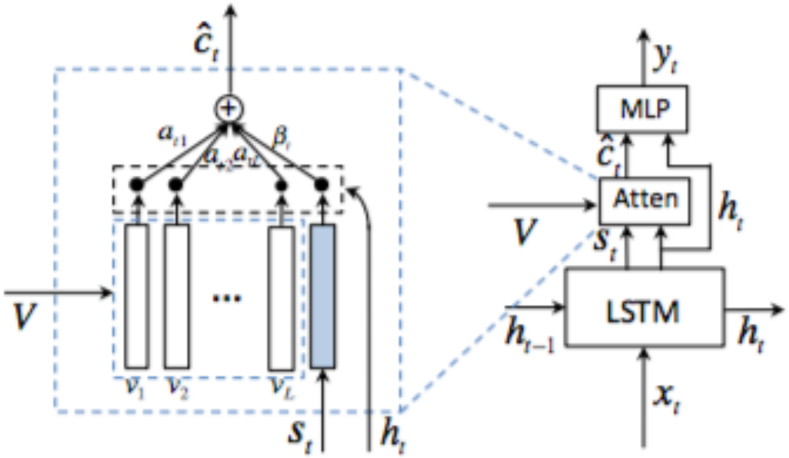
$$\boldsymbol{g}_t = \sigma\left(\boldsymbol{W}_x \boldsymbol{x}_t + \boldsymbol{W}_h \boldsymbol{h}_{t-1}\right)$$

$$\boldsymbol{s}_t = \boldsymbol{g}_t \odot \tanh\left(\boldsymbol{m}_t\right)$$

Network architecture

- Adaptive Attention
  - context vector c_t modeled as mixture of spatially attended image features and visual sentinel
  - sentinel gate β_t decides between attending to the image or the visual sentinel
  - β_t is a scalar in the range [0,1]
  - model adaptively attends to image vs. visual sentinel when generating the next word

$$\hat{\boldsymbol{c}}_t = \beta_t \boldsymbol{s}_t + (1 - \beta_t)\, \boldsymbol{c}_t$$



Datasets

- Flickr30k
  - contains ≈ 32 thousand images collected from Flickr
  - depicts humans performing various activities
  - each image is paired with 5 crowd-sourced captions

- COCO
  - contains ≈ 83 thousand images for training
  - contains multiple objects in the context of complex scenes
  - each image has 5 human annotated captions

Preprocessing

- Images
  - 40 thousand COCO images used
  - 80% for training and 20% for testing

- Captions
  - vocab consists of the 6 thousand most frequent words
  - remove punctuations
  - max length based on longest caption

Implementation

- Encoder-CNN
  - spatial features extracted from last convolutional layer of ResNet-152-V2
  - ResNet was pre-trained on ImageNet
  - spatial features were saved in npy format

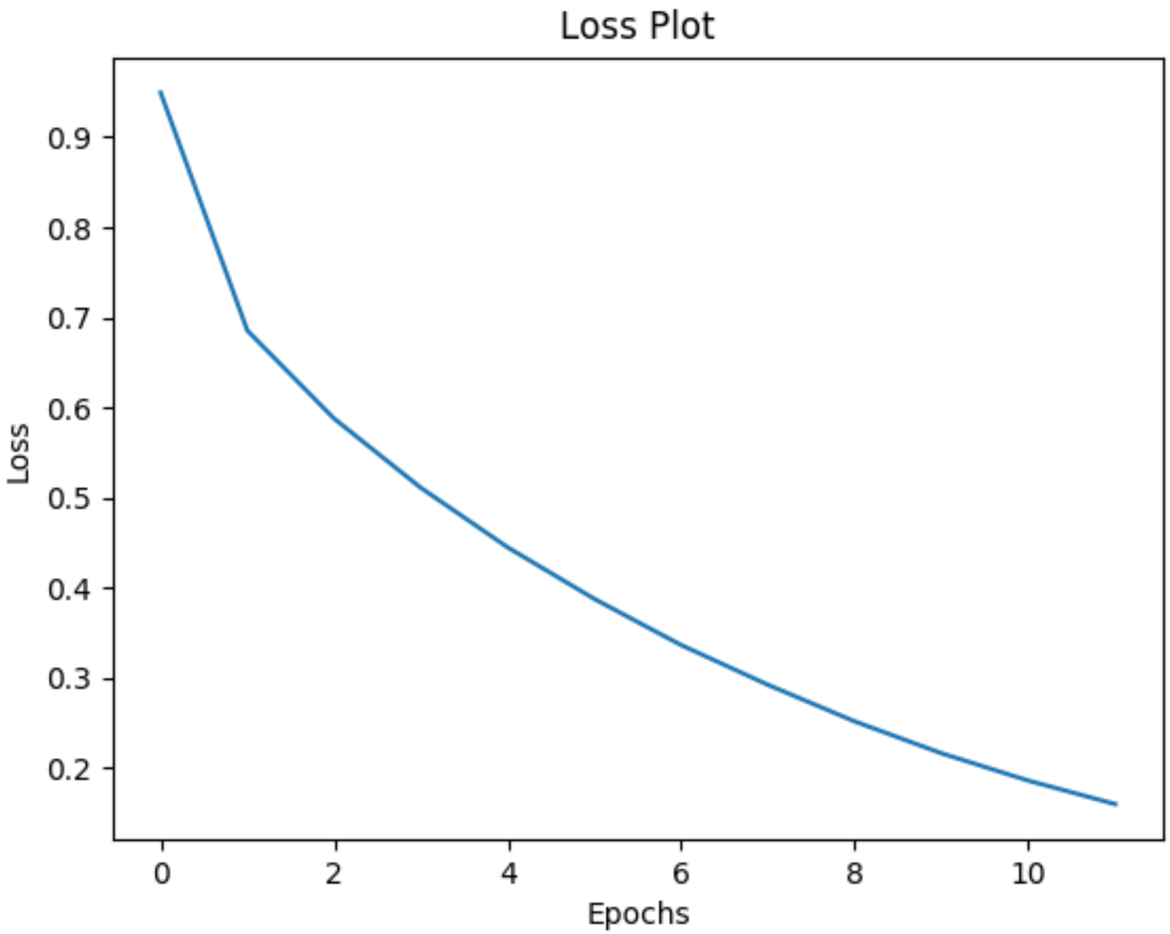
- Decoder-LSTM
  - input is concatenation of word embedding and global image feature

Hidden Size	Embedding Size	Features Shape	grid locations
512	512	2048	49

Batch Size	Epochs
64	12

Results

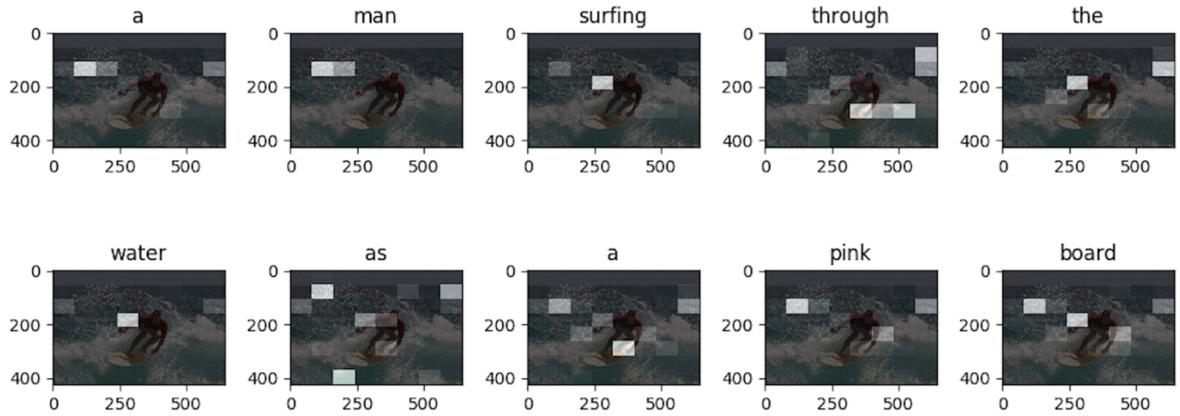
- Spatial Attention



Results

- Spatial Attention

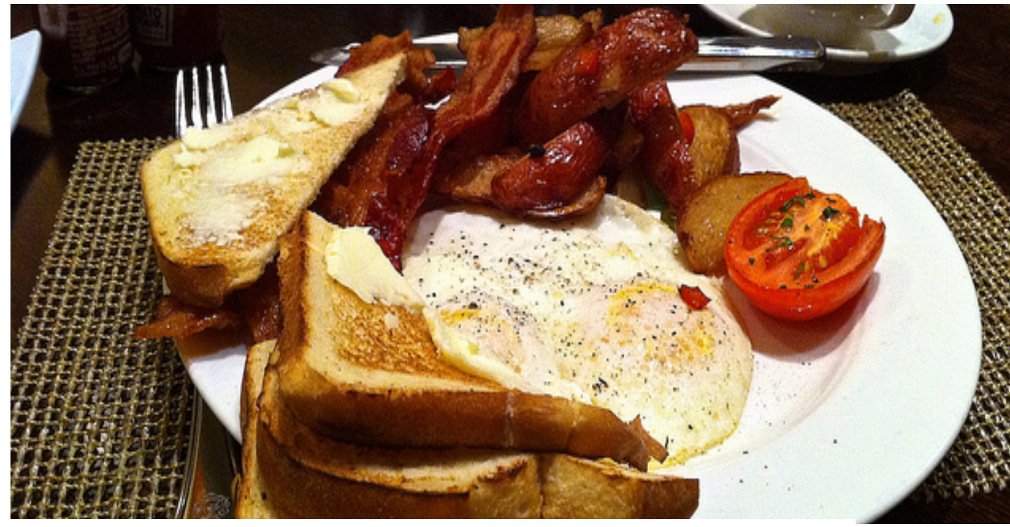
“a man surfing **through** the water **as a** pink board”

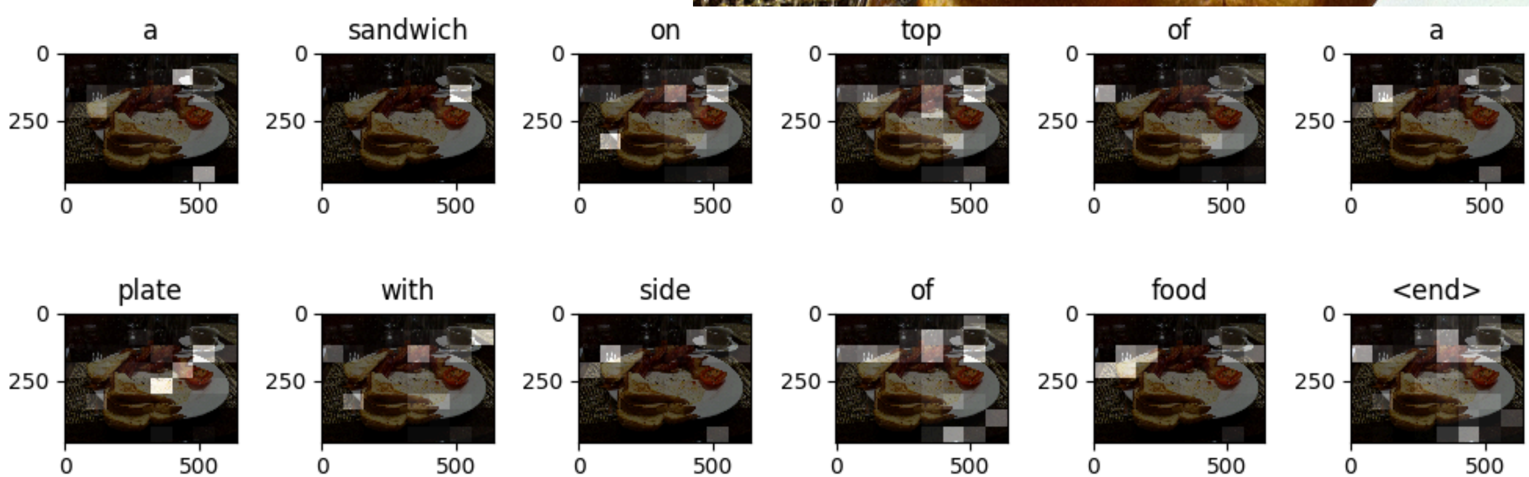


Results

- Spatial Attention

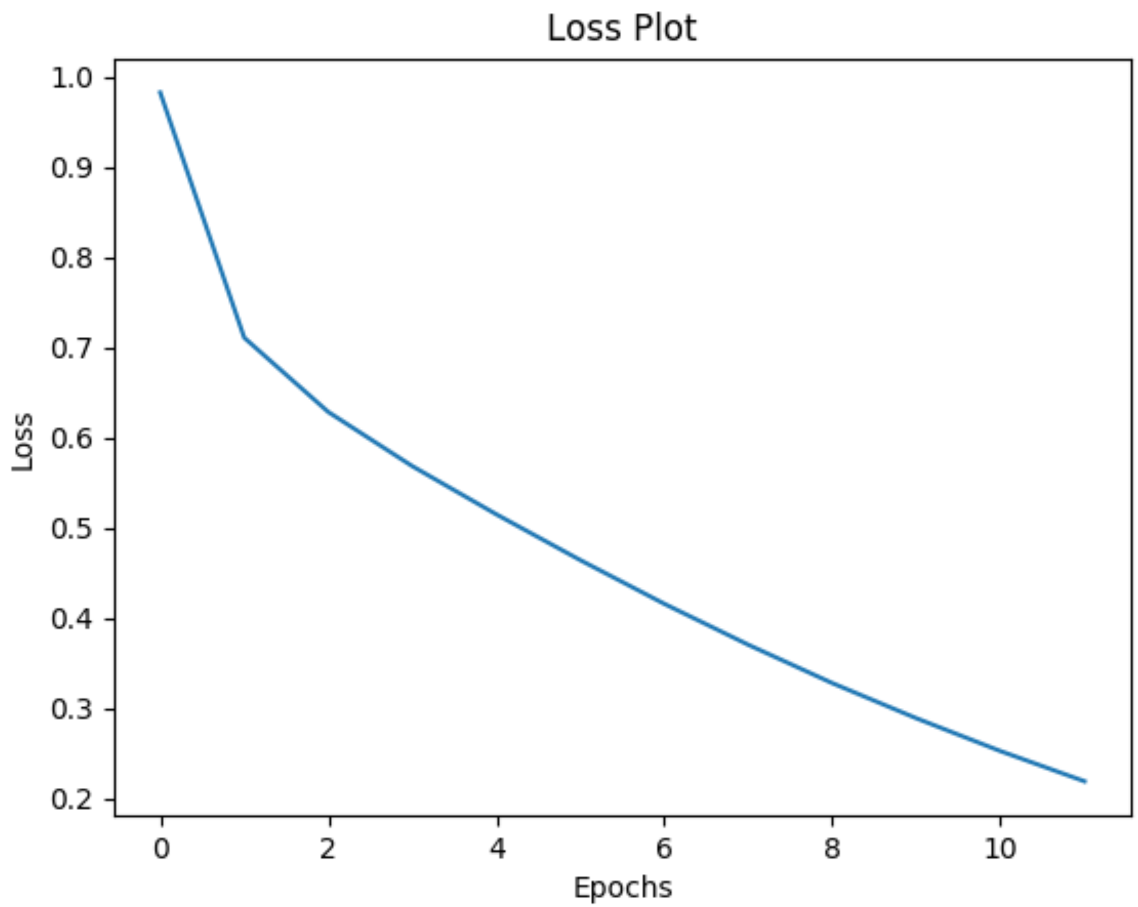
"a sandwich on **top of** a plate **with side of** food"





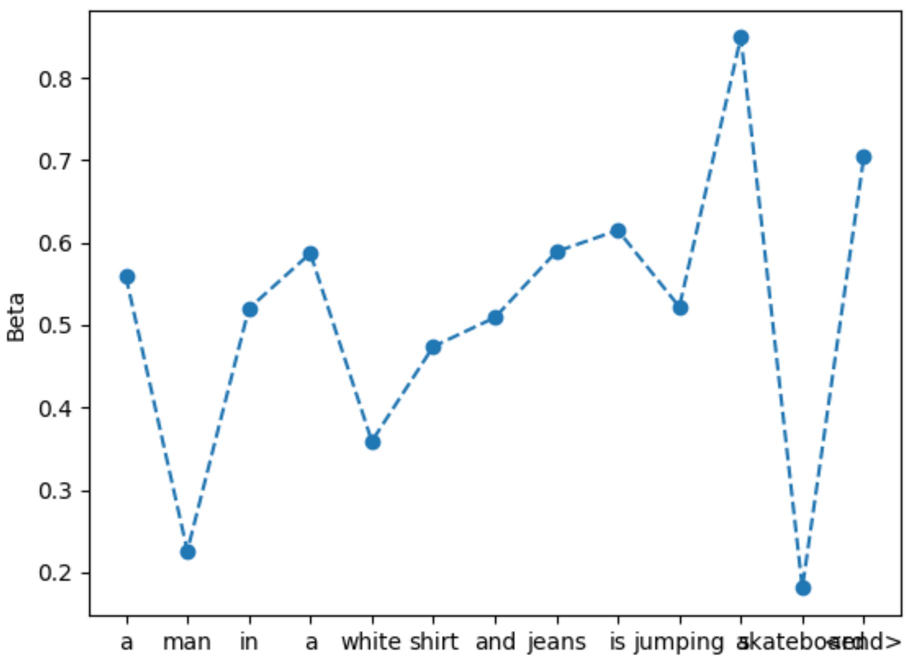
Results

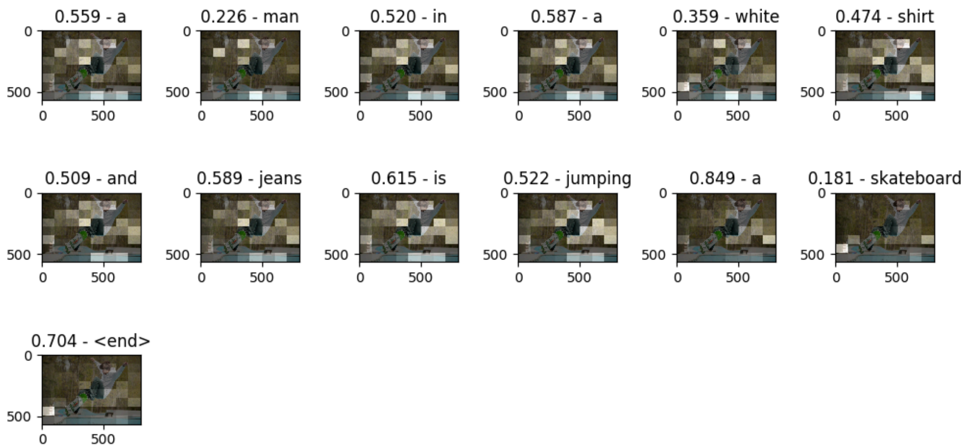
- Adaptive Attention



Results

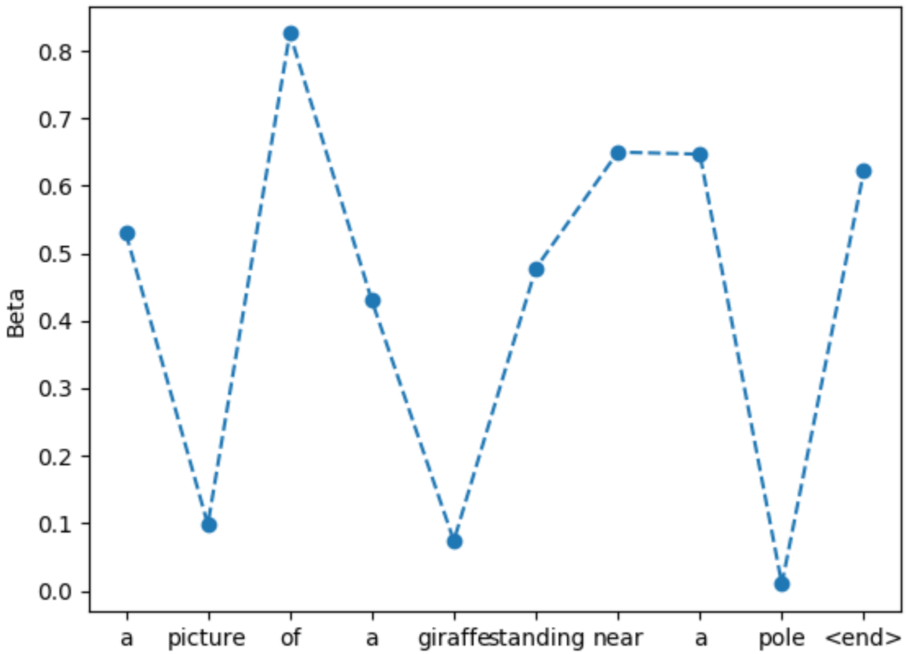
- Adaptive Attention

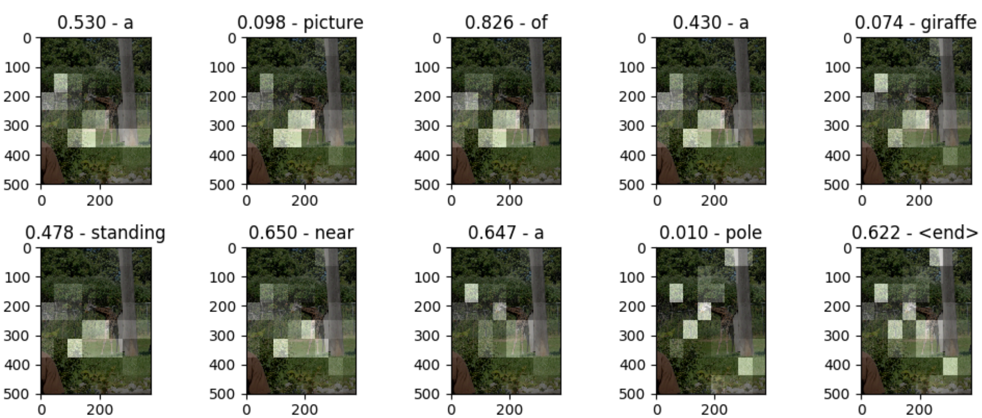




Results

- Adaptive Attention





Future Work

- use contextualized embeddings (e.g., BERT)
- train on entire COCO dataset

Conclusion

- Adaptive attention best performer
- model leans to attend less for non-visual words and attend more for the visual words.

Thank you for your attention