Wikipedia Image Captioning

Hatice Billur Engin Aras bengin@stanford.edu
Kasha Khashayar Akrami kakrami@stanford.edu
Ethan Cheng ethanlc2@stanford.edu
Swathi Gangaraju swathig1@stanford.edu

1.Introduction:

We increasingly rely on online images for knowledge sharing. However, even the most established websites are missing metadata to pair with their images. Current solutions on image captioning rely on simple methods that have limited coverage such as translations or page interlinks. On the other hand, even the most advanced computer vision algorithms are not suitable for images with complex semantics. We aim to build a model that automatically retrieves the English text closest to a Wikipedia image as an attempt to solve this problem.

2.Literature Review:

Image Captioning is a popular research area in Artificial Intelligence. In his review Hossain et al. [1] lists many approaches implemented so far in this field. Earlier attempts treat this problem as a classification task [2]. They work well for specific tasks utilizing hand-crafted features, however, fail to capture the semantic interpretation when there is a need for feature extraction from large and diverse datasets. With the rise of deep learning techniques, features could be learnt programmatically. For example, Convolutional Neural Networks (CNN) [3] became widely used for feature learning, and classifiers such as SoftMax started to be utilized frequently.

Early approaches for caption generation can be categorized into two. The ***template-based approaches*** are fixed templates with several blank slots to generate captions. For example, Li et al. [4] extract the phrases related to detected objects, attributes, and their relationships. The ***retrieval-based approaches*** describe images by retrieving pre-existing captions from a repository. Those approaches cannot generate image specific and semantically correct captions [5]. Contemporary Image captioning methods can be categorized either as ***simple Encoder-Decoder architecture*** or ***Compositional Architecture-Based Image captioning***.

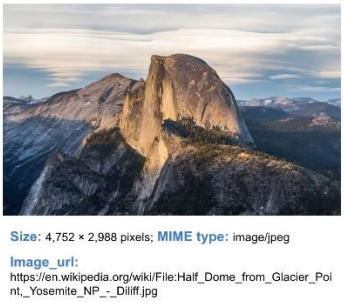
Compositional architecture-based methods are composed of several independent functional building blocks: First, a CNN (such as RESNET, VGG, and Inception) is used to extract the semantic concepts from the image and to provide image embeddings to be input into decoder. Then a language model is used to generate a set of candidate captions. In generating the final caption, these candidate captions are re-ranked using a deep multimodal similarity model. Fang et al. [6] introduced generation-based image captioning. However, these methods are unable to analyze the image over time while they generate the descriptions for the image. In addition to this, the methods do not consider the spatial aspects of the image that are relevant to the parts of the image captions.

Compared to this, in the simple Encoder-Decoder network, global image features are extracted from the hidden activations of CNN and then fed them into a RNN (Recurrent Neural Network) to generate a sequence of words. Based on our research, most of the image captioning methods use Long Short-Term Memory (LSTM) as language model which is the reason behind us choosing it in our baseline approach. It is simpler compared to BERT (Bidirectional Encoder Representations from Transformers) and later models. In the LSTM model, the next word is generated based on the current time step and the previous hidden state. This process continues until it gets the end token of the sentence. Since image information is fed

only at the beginning of the process, it may face vanishing gradient problems. LSTM based models are slowly getting replaced by Bidirectional Encoder Representations from Transformers (BERT) [16] in sequence-to-sequence learning tasks for this reason. Also, Attention based mechanisms are becoming increasingly popular in deep learning because they can address these limitations. They can dynamically focus on the various parts of the input image while the output sequences are being produced. Xu et al. [10] were the first to introduce an attention-based image captioning method.

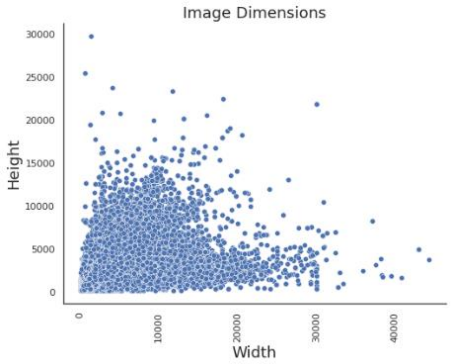
3.Dataset

3.1.Starting Dataset:



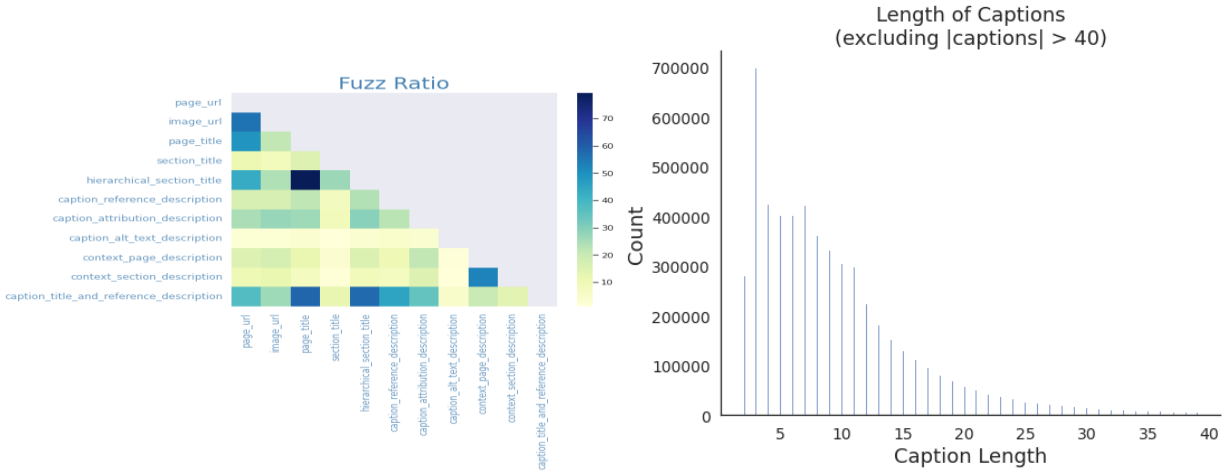
We are using the English language specific subset of the Wikipedia-based Image Text (WIT, kaggle dataset link) Dataset. Input attributes in the dataset are" Image"," Page URL"," Image URL", "Page Title"," Page Description"," Section Title"," Hierarchical Section Title"," Caption Reference Description"," Caption Attribution Description"," Caption Alt Text Description"," Image Type"," Image Height", "Image Width"," Is Main Image"," Context Page Description"," Context Section Description" and Caption Title and Reference Description."

3.2.Exploratory Data Analysis:



Our data consists of ~5.5M data points. 86% of the images are in the JPEG format and the rest of the formats are gif, png, webp and svg + xml. Images have varying heights and widths with a mean height of ~1500 and a mean width of ~1800. Lastly, 30% of the images are main images of the Wiki entries. Our captions have ~1.3M unique tokens and on average each token is observed 5 times among all captions of the ~5.5M images. The frequency of the tokens in our caption vocabulary ranges from 1 to 5.5M. The most common tokens are ‘]’, ‘[’, ‘sep’, ‘of’, ‘the’, ‘,’ ,‘in’, ‘.’, ‘)’, ‘(‘, ‘and’, ‘a’, ‘list’, etc.

The Length of Captions has a right-skewed distribution (mean 9.7, median 8 tokens) ranging between 2 - 962 tokens.



Our objective is to predict the “caption title and reference description” (target value). As shown in the heatmap above (the string similarity of columns obtained via rapidfuzz), there are additional text columns like the target column such as page title, hierarchical section title, caption reference description, etc. As these columns contain similar length text and content to the target column, Fuzz Ratio plot shows a high correlation. However, the “context_section_description” column have additional related vocabulary, as well as target column like content. On further analysis, to maximize the variety of our vocabulary, we decided to utilize the longer (and less similar) “context_section_description” column together with our target column in our main approach.

3.3.Lean Dataset for Training Purposes:

The 5.5M image data required storage and compute power beyond our scope, thus we selected a random subset of 25,443 images for training. To predict the “caption title and reference description,” we have only used images in the baseline approach and included “context_section_description” features in our main approach.

3.4. Test Dataset: We used 25 random examples from WIT dataset as the test dataset for the baseline. For the main approach we used 7500 randomly selected images for test dataset from WIT that are not included in the training dataset. We used WEmbSim score for evaluating how model is doing on the test dataset in comparison to train dataset.

4.Baseline Model

Our baseline image captioning model is an encoder-decoder framework. The encoder part is a CNN (a pre-trained Inception_v3 CNN), as they can produce a rich representation of the input image by embedding it into a fixed-length vector.

In the decoder part, the framework enters the word vector expression into the Recurrent Neural Network (RNN) model. For each word, it is first represented by a one-hot vector, and then through the word embedding model, it becomes the same dimension as the image feature. We have used 4 layered LSTM (512 embedding size, 1024 size of hidden layers) for the decoder part. We chose LSTM as it solves the vanishing gradient problem and the limited memory problem of ordinary RNNs (Recurrent Neural Network) (Recurrent Neural Network). It is the most used decoder method for image captioning task.

WIT database contained URLs to images. We downloaded the images, resized them to 356 x 356 and random cropped them to 299 x 299. We fed the transformed images to the encoder; the encoder generated an embedding vector of size 256. These embeddings are used as input to decoder. The decoder output a caption of maximum 50 words per image.

We have used the Cross Entropy Loss and both the encoder, and the decoder models are optimized by the Adam optimizer. We created vocab dictionary based on word appearing in captions at least once and trained the model for 70 epochs.

5.Main approach

For our main approach, we resized, random cropped and normalized 25443 randomly selected English language images from the WIT dataset like image pre-processing for baseline approach. We continued with Inception V3 CNN architecture to generate the vector representation of the image. The Inception V3 is a deep learning model based on Convolutional Neural Networks, which is used for image classification. It has 42 layers. A convolution layer is the simple application of a filter to an input that results in an activation. Repeated application of the same filter to an input result in a map of activations called a feature map, indicating the locations and strength of a detected feature in an input, such as an image. Pooling layers provide an approach to down sampling feature maps by summarizing the presence of features in patches of the feature map. Dropout is a technique to fight overfitting and improve neural network generalization.

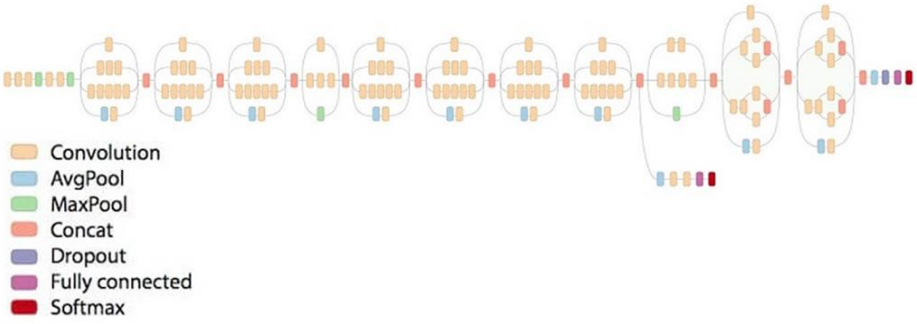


Figure- Inception V3 Architecture

We dropped the last SoftMax layer of CNN. Output of CNN is a vector representation of the image which is of size 2048X1. We used 0.5 as dropout ratio for CNN.

Plus, we used Bidirectional Encoder Representation from Transformers (BERT), state-of-art algorithm for seq-to-seq inference, to generate word embedding vector for “context_section_description” column. We used BERT small uncased pre-trained model by HuggingFace. BERT’s key technical innovation is applying the bidirectional training of Transformer, a popular attention model, to language modelling. The reason

for choosing BERT is BERT considers the context for each occurrence of a given word. The output of BERT has size 768x1 vector per token. For a caption of length 64 words that would be 768*64= 49152 values, making BERT overpower the CNN in number of features. To solve this problem, we added 2 1-D Convolution layers on top of BERT's final hidden layer. The first reduces the number of channels from 768 to 128 with a kernel size of 5, and the second reduces the number of channels from 128 to 64 with a kernel size of 5. The final output is then flattened to a 64x(64-8) =3584 dimensional vector. This results in BERT output of approximately 3584 size vector per input text, comparable in size to CNN output. Any text greater than 64 words is truncated to 64 tokens which was chosen arbitrarily since the average number of tokens is around 61. We also used a dropout ratio of 0.5 for BERT output as well.

The output of CNN and BERT is combined using torch.concat and is input to another linear layer to adjust the dimensionality from (2048 + 3584) to embed_size, which is a hyperparameter. Output of linear layer is input to decoder LSTM to generate predicted captions. We used embed size=1024, hidden layer size=1536 and number of layers=4 as hyperparameters to tune.

We used WEmbSim score to evaluate the model. Given a URL of the input image together with the "context_section_description" column, we predicted five captions.

Example Input	Example Output
URL:'https://upload.wikimedia.org/wikipedia/commons/2/28/Deer_Park_Wisconsin_Downtown_WIS46.jpg' **context_section_description:** 'Deer Park is a village in St. Croix County, Wisconsin, United States. The population was 216 at the 2010 census.'	• Deer Park Wisconsin • Downtown Deer Park • Deer Park Village • Deer Park Main Street • The village downtown
URL:'https://upload.wikimedia.org/wikipedia/commons/3/3d/The_Flintstones.png **context_section_description:** 'The Flintstones was an animated sitcom directed by William Hanna and Joseph Barbera. There were six seasons consisting of 166 episodes. Each episode was 25 minutes long. A pilot episode was released in 1959 and the show aired between September 30, 1960, and April 1, 1966.'	• Seth MacFarlane's logo • The Flinstones' logo • The Flinstones • The logo • The red Flinstones logo

6. Evaluation Metric

WEmbSim[11] is a cosine similarity-based measure which uses the mean of word embeddings for caption evaluation. WEmbSim is defined mathematically as follows. For a set of reference captions R, and a candidate caption to be evaluated C = [w1, w2, . . . , wn], we define the function v˜(·) which maps a caption to a vector via the mean of word embeddings representation using the embedding matrix V. We used mean as the combining function.

$$\bar{v}(C) = \frac{1}{n} \sum_{\forall w_i \in C} V_{:,w_i} \quad (1)$$

Using this, and the cosine similarity function:

$$\text{cossim}\left(\bar{a}, \bar{b}\right) = \frac{|\bar{a} \cdot \bar{b}|}{|\bar{a}||\bar{b}|} \quad (2)$$

We define the scoring function as:

$$\text{Score}(C \mid \mathcal{R}) = \underset{\forall R_i \in \mathcal{R}}{\text{Combine}} \text{cossim}\left(\bar{v}(C), \bar{v}(R_i)\right) \quad (3)$$

We used glove-wiki-gigaword-100 pre-trained word embeddings vectors to generate vectors for captions. We calculated cosine similarity score between vectors for actual caption in the database to predicted caption for the image. We used the average of the similarity scores of the 5 predicted captions to evaluate the predictions. Then we average all the similarity scores for all the images in the training or test set to get the WEmbSim score for the model.

7. Results & Analysis

We got WEmbSim score of **.614** on 1000 randomly chosen samples from the training set using Baseline approach. We created an Oracle dataset of 20 images where each of the 4 project team members predicted 5 captions for 5 images each. The WEmbSim score comparing our human-generated captions to Wiki captions is **.617**. The WEmbSim score comparing model generated captions to the Wiki+Human-generated captions on the Oracle is **0.595**. The Oracle dataset is our test dataset for baseline approach. Initially we tried to run our baseline model on the entire database and ran into memory and processing speed issues. We downloaded a dataset of 25443 images and input this to our model described in the main approach. We trained the model for 170 epochs. Compared to the baseline approach the scores of the main approach appear to be slightly lower. Also, model performance appears comparable to Oracle set that included human generated captions. The interpretation of the scores and possible reasons are included in the 'Error Analysis' section.

	Training Dataset	Test Dataset
Baseline Approach	0.614	0.617
Main Approach	0.595	0.571

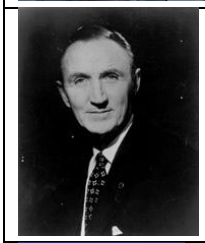
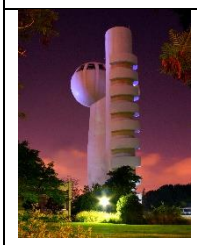
On further analysis of the score and predicted captions, we realized that we did not apply the dropout ratio on BERT as planned. This would explain why some predicted captions matched input captions 100% while others did quiet poorly. We fixed the issue. Model is still re-training at the time of submission. Will post the results in the repo once re-training completes.

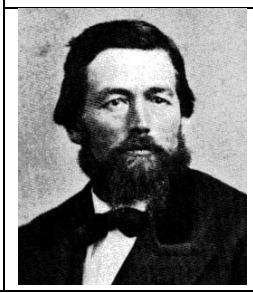
Model did perform better than our expectations and predicted comparable captions exceeding limits of available memory and processing capabilities. Initially, most of the captions were US-centric in content. After epoch 49, we started observing diversity in the identification of landmarks, and other regional attributes in the predicted captions. This could be due to using English language only image metadata from Wikipedia, majority of which may be US related. As we only used images for the baseline, we noticed the model is having difficulty in predicting context-specific captions and captions for complex images like names of people in the image. One idea to address this was to include contextual text in section description field to help the model learn the context. We incorporated this idea in our main approach. Contrary to our assumption, our main approach did similar to baseline in this aspect. This could be due to low frequency of the same name or place captions for model to learn enough. We scored the similarity of predicted captions to ground truth in train, and test datasets.

Below table provides 2 training set examples with generated captions using baseline approach with WebSim scores and comments.

Examples	**Expected caption**	**Generated caption**	**Comment**	**WEmbSim Metric**
	<SOS> skra be_chat_w [sep] first six during a match of plusliga with asseco resovia at atlas arena, __d_ on 30 november 2014. <EOS>	<SOS> list of international goals scored by wayne rooney [sep] rooney being tackled by the united - time award in 2014 <EOS>	Model actual recognizes a sport being played, mentions goals being score	Similarity: 0.907
	<SOS> patricia neal [sep] patricia neal at the tribeca film festival (2007) <EOS>	< SOS> list of female ministers of the United States: Africa [sep] <EOS>	Detected a 'female' person correctly	Similarity: 0.772

8. Error Analysis

Category	Examples	Expected caption	Generated caption	Comment	WEmbSim Metric
Good		list of protected heritage sites in huy [sep]	list of protected heritage sites in liège [sep]	Recognizes site accurately but not 100% accurate on the name	0.972
		list of museums in the republic of ireland [sep]	list of museums in northern ireland [sep]	Detected the building is a museum accurately	0.9711
		1966 united states senate elections [sep]	2008 united states senate election in montana [sep]	Detected the country and united state election correctly	0.967
		list of tallest buildings in baltimore [sep]	list of tallest buildings in british columbia [sep]	Detects tallest building but not the place	0.9556
So-So		staphylococcus aureus [sep] s. aureus on trypticase soy agar : the strain is producing a yellow pigment staphyloxanthin	list of acacia species known to contain psychoactive alkaloids [sep]	This is understandable as acacia flowers do have yellow pigment	0.7480
		<SOS> economy of israel [sep] weizmann institute of science , rehovot <EOS>	<SOS> list of star wars planets and moons [sep] map of the star wars galaxy (legends) <EOS>	Does identify the sky. Generates understandable caption based on colors and shapes in the image. Building does look like spaceship!	0.667

		georg dohrn [sep] hall of the konzerthaus breslau	list of paintings by jacob van ruisdael [sep]	This is understandable. Due to the color scale of the image, it looks like a painting	0.6045
		<SOS> rhododendron sect . vireya [sep] rhododendron ' kamrau bay ', a hybrid whose background includes r. zoelleri and r. laetum[3] <EOS>	<SOS> list of carnivorous plants [sep] stylidium bulbiferum <EOS>	The model understands that this is a plant, however, cannot capture the plant name. The color of flower of plant in generated caption is like one in image	0.6050
		austrian walled towns [sep]	<SOS> st patrick 's basilica, waimate [sep] <UNK> 's basilica, waimate , interior <EOS>	Detects that it is place with a structure. Does not get the name of the place right	0.466
		fort amiel museum [sep]	list of united states commemorative coins and medals (2010s) [sep]	This is unrelated except many coin images do have these colors and hue	0.3968
Bad		granville henderson oury	charles n. arnold	Detects that it is name of a man	0.3865

Terrible		marsha shandur [sep] shandur in 2006	norah runge [sep]	Detects a female name which is better than what score indicates. Interesting male image is scored higher than female image where in both cases model identified wrong names	-0.0567
		luigi rossini [sep]	list of united states commemorative coins and medals (2010s) [sep]	This is unrelated except for colors and hues. Looks like it is not learning enough details in the image to know negative examples of coins and medals	-0.019452

Table depicts a sample set of captions that were generated by our models, which we have attempted to sort qualitatively. We arrived at the definitions based on approach used in Neural Image Captioning paper by Lakshay Sharma et al. [17]

Good caption to be one that is an accurate and illustrative description of what is happening in the picture. For example, the model was able to predict 'tallest building' as indicated in the input caption.

A so-so caption is a description close to the ground truth (i.e., what is really happening in the picture) but missing or misreading intricacies, or with incorrect grammatical/semantic constructs. For example: Model identified image focus object as plant species and of same color.

A bad caption is an inaccurate but somewhat understandable caption of the image. Bad captions include mistakes such as gender, object misclassification, or relational errors. In the first one, the predicted image caption includes 'coins and medals' mis-classifies object but kind of makes sense on color and hues; in the second one model identified that it is a person and predicted right gender in the image but got the name wrong.

A terrible caption contains one or more errors that can lead to complete misrepresentation of what is happening in the picture or the inability to form a complete caption. In the image about water stream, model classifies it as coins and medals.

Overall, the model is struggling to recognize names of people and places accurately, however, it classifies focus objects in the image well. One reason for model was not doing well on some of the

captions could be that the pre-trained model we used for word embeddings may not encompass all the vocabular needed. The WEmbSim using glove-wiki-gigaword-100 pre-trained word embeddings is an extremely poor metric of names of people and places and will always be extremely low if the expected caption is a name, and the generated caption is not exactly that name. Also, in all cases we analyzed captions were grammatically correct. We also saw many instances of caption containing word 'list.' One reason may be that because 'list' is one of top 15 most common tokens in dataset, model is learning to use it frequently. Also, many of the expected captions (perhaps due to how Wikipedia is formatted) start with 'list of ...' when the image is not a list itself. Model predicts ''list of united states commemorative coins and medals" 1250 times on the test set. Out of these 1250 default predictions, 1243 of the inputs have an empty context_section_description text. This reliance on the text input signals heavy dependence on the text compared to the image.

9. Conclusion and Future Work

Overall, in this project we were able to create a simple image captioning model that is doing well when compared to human generated captions for Wikipedia images. We were able to pre-process the data, create image and text to vector embeddings and use multimodal input and encoder-decoder architecture using Inception V3 as CNN and BERT as text encoder and LSTM as decoder and generate captions for Wikipedia images. While we did have challenges in getting reasonable captions early in the project, post hyperparameter tuning and running the model for more epochs we noticed improvement in the performance of the model. Based on the challenges we faced and our analysis of the observed results, we propose the following possible improvements for future work on this model

- Execute more tuning with BERT dropout ratio to decrease effect of overfitting
- Experiment with other popular Image Captioning approaches such as attention mechanism which we could not implement due to time constraints.
- Experiment more with feature extraction techniques to include other useful metadata features in the input to ensure model learns semantics and relational aspects better
- Train the model on a larger dataset which we could not do due to resource constraints
- Augmenting the training set with samples that are either missing the image or section text to learn a more robust model that can handle partially missing inputs. This can also be simulated by applying a dropout layer or mask - like BERT's MLM training model - to the input directly as well.

10. Code

https://github.com/sgangaraju1/CS221_project

11. Link to Project Video on Youtube

https://www.youtube.com/watch?v=W-G9_jbbZK4

12. References

1. A Comprehensive Survey of Deep Learning for Image Captioning - Md. Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin and Hamid Laga 2018.

2. Navneet Dalal and Bill Triggs. 2005. Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, Vol. 1. IEEE, 886–893.
3. Yann LeCun, LÃľon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86, 11 (1998), 2278–2324.
4. Siming Li, Girish Kulkarni, Tamara L Berg, Alexander C Berg, and Yejin Choi. 2011. Composing simple image descriptions using web-scale n-grams. In Proceedings of the Fifteenth Conference on Computational Natural Language Learning. Association for Computational Linguistics, 220–228.
5. Yunchao Gong, Liwei Wang, Micah Hodosh, Julia Hockenmaier, and Svetlana Lazebnik. 2014. Improving image sentence embeddings using large weakly annotated photo collections. In European Conference on Computer Vision. Springer, 529–545.
6. Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr DollÃąr, Jianfeng Gao, Xiaodong He, Margaret Mitchell, and John C Platt. 2015. From captions to visual concepts and back. In Proceedings of the IEEE conference on computer vision and pattern recognition. 1473–1482.
7. Sepp Hochreiter and JÃijrgen Schmidhuber. 1997. Long short-term memory. Neural computation 9, 8 (1997), 1735–1780.
8. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In International Conference on Learning Representations (ICLR).
9. Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. In Association for Computational Linguistics. 103–111.
10. Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In International Conference on Machine Learning. 2048–2057.
11. WEmbSim: A Simple yet Effective Metric for Image Captioning - Naeha Sharif, Lyndon White, Mohammed Bennamoun, Wei Liu, Syed Afaq Ali Shah 2020.
12. Image Captioning with Compositional Neural Module Networks - Tian J et al, IJCAI 2019.
13. Improving Image Captioning with Better Use of Caption - Shi Z et al, ACL 2020.
14. Comprehensive Image Captioning via Scene Graph Decomposition - Zhong Y et al, ECCV 2020.
15. Diverse Image Captioning with Context-Object Split Latent Spaces - Mahajan S et al, NeurIPS 2020.
16. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding-Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, May 2019
17. Neural Image Captioning-Lakshay Sharma, Elaina Tan, Jul 2019