Capstone Proposal

Machine Learning Engineer Nanodegree

Build a Social Image Description Platform - Computer Vision

Nilakshan Kunananthaseelan
October 1,2020

Domain Background

Computer vision problems have achieved remarkable results with diverse deep learning algorithms,availability of large datasets and high performance hardware sources.Lot of state of art models has been developed by researchers in the image tagging domain:Describe an image with suitable tags.
In this project,I'm going to develop a deep learning model which is capable of describing an image with most suitable tags.The model is then deployed to a web interface so that a user can upload an image and generate a description using tags.This project has real world application as we seen in social networks like facebook and instagram where user can retrieve images based on tags or suggests tags to user based on their recent activities.Additional,it can be aided with acoustic models to describe images in the social media platform to vision impaired people which could break the barrier to access social media for them.
There are diverse architectures available for image captioning and image tagging problems.Most of them are COCO based model.**"**Show and Tell" by Vinyals,et al. [1] shows good performance with most of image tagging datasets.We can use the results for the Flickr30k dataset as the benchmark."TagProp " by Guillaumin,et al.[2], "Fast Image Tagging" by Chen,et al.[3] and "OSCAR" by Microsoft [4] are other recent works

Problem Statement

We have two problems to solve.First one is to automate the image captioning/tagging with a machine learning model and second one is to host a website which can make use of the trained model and predict the relevant tags for unseen images.

1. Image annotating
   The image annotation can be stated as the process of describing an image with a caption or set of tags.Fig.1 can be described with tags such as 'bridge','sea','road','clouds'.

Figure 1

This task can be characterized as a multi-label classification problem:each instance in the dataset associated with multiple labels as of Fig.1.

From Zhang et.al we can formally define this problem in the following way.

Suppose $\mathcal{X} \subset \mathbb{R}^d$ denotes the d-dimensional space of images and $\mathcal{Y} = \{\ell_1, \ell_2, .., \ell_q\}$ denotes the label space with q possible class labels.Each instance, $x_i \in \mathcal{X}$, of the dataset can be linked with subset of labels from $L_i \subseteq \mathcal{Y}$.We can represent subset of labels in one-vector form with q-dimensional vector: $Y_i = \langle y_1, y_2, ... y_k \rangle$ which ultimately set the task of multi label learning problem as a mapping function $h : \mathcal{X} \rightarrow 2^y$.

2. Website hosting
   The best performed model will be deployed on a website to mae prediction on unseen instances.The website will be mimicking the behaviour of instagram and flicker in which a user can create their account and upload photos.The deployed model will predict the tags/caption relevant to the photos and save them in a database.The user can retrieve the data base on the tags or captions and these images will be exposed to global feed.

Datasets and Inputs

The COCO dataset is generally used in image captioning/tagging model preparation,but considering the size of the dataset and available resource for training,I preferred to work with MIRFLICKR (https://press.liacs.nl/mirflickr/#sec_introduction) dataset since it is open,interesting and has a practical usage in the research community.
The MIRFLICKR-25000 open evaluation project consists of 25000 images downloaded from the social photography site Flickr through its public API coupled with complete manual annotations, pre-computed descriptors and software for bag-of-words based similarity and classification and a matlab-like tool for exploring and classifying imagery.The smallest version contains 25000 instances with 1386 tags.The average number of tags per image is 8.94 in which most of them are in English.Since the dataset is extracted from Flicker and it covers most of the general tags which makes it a potential one to use in this project.

Solution statement

The task of image tagging can be considered as an encoder-decoder problem where image features are encoded and then decoded with relevant tags. Tant,et.al.[5][6] suggested a similar approach:'merge-model' .The problem domain can be divided into two separate modules,one is an image-based model [7][8] which extracts features out of an image,and other one is language-based model [9][10])which is capable of translating the feature vector into tags.

Benchmark model

"Show and Tell: A Neural Image Caption Generator" by Vinyals,et al shows good performance with most of image tagging datasets.We can use the results for the Flickr30k dataset as the benchmark."TagProp: Discriminative Metric Learning in Nearest Neighbor Models for Image Auto-Annotation" by Guillaumin,et al., "Fast Image Tagging" by Chen,et al. and "OSCAR" by Microsoft are other recent works that can be considered to evaluate performance of the model.

Evaluation metrics

The general metric used to measure the relevance of captions or tags related to the images is Bilingual Evaluation Understudy (BLEU) score.F1-score,Precision,Recall and Area-under-curve(AUC) are the other metrics that can be used to evaluate the model.

Project design

The road map of the project will be as follows:

1. Literature Review

   Read some recent works in image annotation/image captioning to get some idea on flow of the work
2. Data Collection

   To train and evaluate the model,we need a suitable dataset.Instead of creating a dataset by collecting images from a website,it is reasonable to use an open source dataset.
3. Data Understanding

   We need to understand the data clearly before making further steps.Without clear knowledge of the dataset,we can't effectively reflect the model's purpose and results will be vague.
4. Data Cleaning

   Often the online datasets contains lot of unnecessary data.Using them inside the model is not efficient in prediction and time consuming.It is essential to clean the data so that it is relevant and precise.For example incase of text data associated with images,we might need to convert to lowercase,remove punctuation if any,remove numbers of any.
5. Data Preprocessing (Images and Tags)

   Images : It has been planned to use transfer learning for feature extraction to feed the final model.We need to convert the image dataset to a valid size to feed to the extractor.For example we need to resize images to 299x299xx3 sized tensor if we wish to use InceptionV3 architecture

   Tags : After cleaning tags,we can create a tag list containing a unique set of tags available in the dataset.Additionally,to make the model to behave more robust to outliers we can set a threshold for most occurring tags.We each image we can create one hot vector of tags,1 if a particular tags belong to the image 0 if not.

6. Data Preparation for training

   We can use generators or data loaders to prepare the data in a way it will be convenient to be given as input to a deep learning model.We could represent each training instance with an image vector and associate tag-one hot vector.
7. Word Embeddings

   Similar to discussed in data preprocessing,word embedding can be done in the form of one-hot vector representing the membership of each unique tag.
8. Developing Model Architecture

   Image based model : In the encoder part,we will use a pretrained model to extract feature vector,get the output from final fully connected layer before softmax function merge with an RNN model (LSTM or Transformer)

Text base model : For decoding we would engage a RNN model which is then merged with an image based model and create a feed forward network with softmax function to predict the tags associated with the image.

9. Inference

   After training the model,do testing with images and analyze the predicted result

10. Evaluation

    We can use the test set to evaluate the model's performance on the dataset using standard metrics so that we can get an idea how good the model is.

11. Developing a website

    Develop a website with flask technology which has UI to upload images to datastore,show the predicted results,search options to retrieve.

12. Deploy the best model on the website

    We can get different models by hyper parameter tuning and different data representation,the best model will be deployed on the developed website.

13. Run the website in localhost

    Run the website so that a user can upload an image and get the tags predicted and search an image based on tags.

14. Upload pictures and save the predicted results

    The ultimate goal of the project.

References

[1] Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan.**"***Show and Tell: A Neural Image Caption Generator"* [link]

[2] Matthieu Guillaumin, Thomas Mensink, Jakob Verbeek and Cordelia Schmid. *"TagProp: Discriminative Metric Learning in Nearest Neighbor Models for Image Auto-Annotation"*[link]

[3] Minmin Chen, Alice Zheng, Kilian Weinberger."*Fast Image Tagging*" [link]

[4]Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu,Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei,Yejin Choi, and Jianfeng Gao.*"Oscar: Object-Semantics Aligned Pre-training*
for Vision-Language Tasks"[link]

[5] Marc Tanti, Albert Gatt, Kenneth P. Camilleri."*What is the Role of Recurrent Neural Networks(RNN) in an Image Caption Generator?"* [link]

[6]Marc Tanti , Albert Gatt , Kenneth P. Camilleri ."*Where to put the Image in an Image Caption Generator"*[link]

[7]Kaiming He,Xiangyu Zhang,Shaoqing Ren,Jian Sun."*Deep Residual Learning for Image Recognition"*[link]

[8]Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna."*Rethinking the Inception Architecture for Computer Vision"*[link]

[9]Sepp Hochreiter,Jurgen Schmidhuber. "Long short term memory" [link]

[10]Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin."*Attention is All You Need"*[link]