Image-Caption Mismatch Classification with Context

DARPA SemaFor UC Berkeley TA1 team

Evaluation 2 self-evaluation submission

Problem description:

We see the potential threat as follows: potential attacker wants to push fake news through social media, and to make the statement more convincing, the attacker finds an image to support it. The image originally has another caption and is not related to the statement the attacker is trying to support.

Thus the problem is to detect image-caption inconsistencies in news posts. We view it as a binary classification problem.

Approach description:

In our approach we utilize additional information. We are trying to mimic the approach of a human fact checker to social media posts. We assume that, having a social media post (image + caption pair), a human fact checker runs a reverse image search on the image and finds another caption for it. Then, by matching the content of the original caption and the found caption the fact checker decides if the original caption was consistent with the found caption or not. Thus, our input data consists of 3 items: caption 1 (original), caption 2 (found) and image. We assume that caption 2 is always consistent with an image. Each data point has a binary label: 0 - if caption 1 is consistent with an image and 1 if image and caption 1 are not consistent.

Example data point:

***Caption 1 (Original)**:* Andy Yates, a political consultant, testified on Tuesday at a hearing on North Carolina's Ninth Congressional District race, which remains undecided.

***Caption 2 (Found)**:* Andy Yates, a political consultant with Red Dome Group, testifies during the second day of a public evidentiary hearing on the 9th Congressional District voting irregularities investigation Tuesday, Feb. 19, 2019, at the North Carolina State Bar in Raleigh.

***Label:** 0 (Consistent)*

Finding real data for this problem is a very difficult and time-consuming process and we believe that good quality fake news dataset doesn't exist. We artificially construct the dataset from the COSMOS dataset. Train and validation splits of this dataset have identical structure: images from news articles and multiple captions obtained for those images by running reverse image search, so that more than a half of images have multiple captions associated with it. Test split of the COSMOS dataset was meant to be real fake samples, but careful exploration has shown that quality of the data in this split is not sufficient.

During construction of our dataset we rely on the CLIP model: neural network model which is aimed to embed images and texts into the same space. We use CLIP similarity score, which is roughly equal to inner product of two features vectors extracted by CLIP model To construct consistent points (label 0) points we select caption pairs from the COSMOS set directly such that CLIP similarity score between two captions is inside an interval [0.35,0.95]. Each pair is included twice with two different order of captions. To construct inconsistent samples we use a technique which is similar to the technique used in NewsCLIPpings dataset. We adversarially choose inconsistent caption 1 from the dataset, the choice criteria is a product of two CLIP similarity scores: caption 1 and caption 2 similarity score and caption 1 and image similarity score. We have 103 thousand correct samples and 103 thousand fake samples.

Then we split the dataset into training, validation and test partitions, and then train a model to achieve high accuracy on the test set, which model wasn't exposed to during the training. We use the CLIP model to extract features and apply a 3 layers MLP classifier on top of the vector of concatenated features. Weights of both CLIP model and classifier are trained.

Content list and description:

Eval_model.py - script to evaluate a trained model
Train_model.py - script to train model from the data
models/ - folder containing trained models
    img_cap1_cap2_paired_best.pth - base model
    img_cap1_best.pth - model which is trained having access only to caption 1
Failure_cases_img_cap1_cap2.html - html document with model failure cases
Dockerfile - docker file to build the docker container
download_dataset.sh - script to download original COSMOS data
datapoints_from_COSMOS2_real.pth,
datapoints_from_COSMOS2_fake.pth - data points for our dataset

Running the scoring script:

1) Download the data:
   We use the COSMOS dataset as the basis. To download the dataset, please run download_dataset.sh via terminal. The script is provided by Shivanji Aneja, one of the authors of the COSMOS dataset paper.

2) Build docker image from .Dockerfile
3) Run docker image, mount the the folder where you download the dataset (step 1) into /scratch/aam/COSMOS and the folder containing downloaded code into some empty directory (‘/home’ works). Example command:

   **docker run -dit -p 8873:8873 **
   **-v /research/aam/COS_CLIP_v23:/home **
   **-v /scratch/aam/COSMOS:/scratch/aam/COSMOS **
   **--gpus all --rm --shm-size=16g **
   cuda_conda_clip

4) Enter the docker container and execute

   python Eval_test.py.

   Additional arguments might be passed to run the model on image + caption 1 data only. Example:

   **python Eval_model.py --mode img_cap1 **
   --model_path models/img_cap1_best.pth

Analytics results:

Evaluating the model on a test set achieves 82% accuracy and ROC AUC score 0.89 (figure of ROC curve might be found in the same folder, named ‘*result_figuern.jpg*’, where *n* is an experiment number). Observation shows that some of the model failure cases are actually hard: inconsistent samples are in-fact consistent or don't have sufficient information; consistent samples have two captions that are not very related, even being captions to the same image from two different web-pages.

Additionally, we run the same experiment on the model which has an access only to caption 1 and image. This case is similar to the NewsCLIPpings problem, and we achieve 65% accuracy, which is very close to the result reported in the NewsCLIPpings paper. Thus, we demonstrate that having access to the probe caption boosts the results significantly.

Please find examples of model failure cases below:

Example of incorrectly predicted inconsistent sample:

Predicted logit is: *-21.06*
Caption 1: *Serbian fans clash with riot police at last year's match*
Caption 2: *Fans of Serbia roll a barrel towards the riot police during clashes at the Euro 2016 Group I qualifying soccer match between Serbia and Albania at the FK Partizan stadium in Belgrade October 14, 2014.*

Example of incorrectly predicted consistent sample:

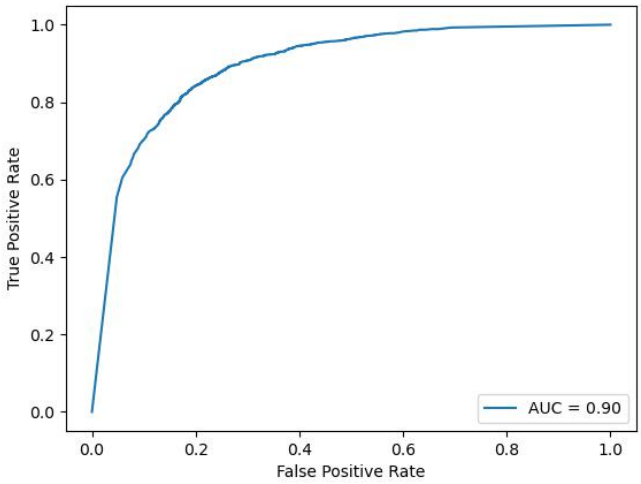
Predicted logit is: *16.27*
Caption 1: *Brazil, Jair Bolsonaro is the new president*
Caption 2: *Brazil, revolt in the penitentiary: a carnage*

Result scores:

Accuracy: >82%

ROC AUC score: >0.89

ROC Curve:



Description of scoring metrics:

Standard Accuracy and ROC AUC score metrics.