Image Captioning for Social Media Post

Joanne Jiang, Zhongyao Ma, Yangyiying Pu, Chengwu Shen, Huidi Wang

rarepups • Follow
New York, New York

rarepups #maltese #maltesepuppy #maltesedog #malteselove #maltesepuppies #malteseofficial #malteselovers #poodle #poodlepuppy #poodlepuppies #dog #dogs #dogsofinstagram #pupsofinstagram #pupstagram #dogstagram #dogsofig #puppiesofig #puppy #puppies #puppiesofinstagram #pomeranian #pomeranianpuppy #instadog #puppylove #sandiego #newyork #brooklyn #sanfrancisco #losangeles

4h Reply

sheilabernie Definitely 10

Liked by aasmari_ and others

Project Motivation

- Encourage users to add more popular and interesting tags on their posts
- Increase user engagements i.e. likes / comments / follows and better search results
- Positive business impacts of social network services

Project Description

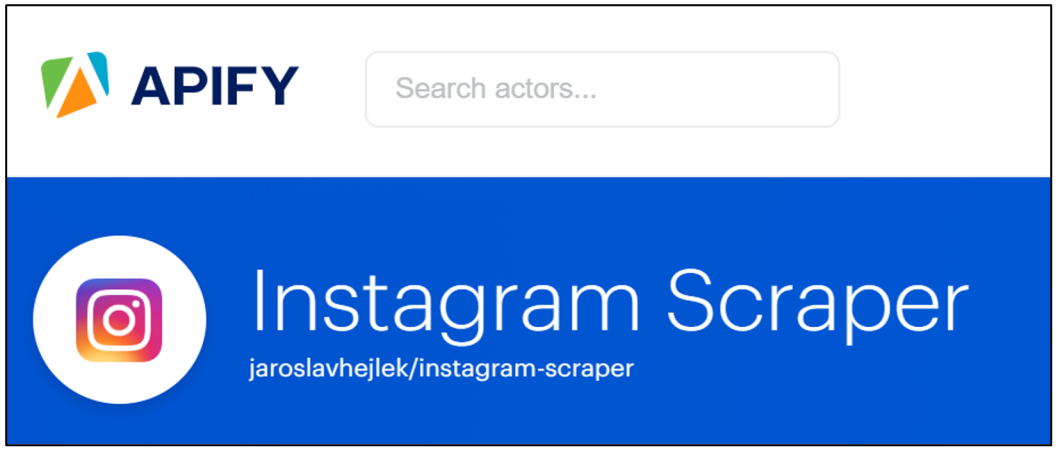
- **Target**: SNS (Facebook, Instagram, Pinterest) & end users
- **Objective**: Improve user experience by generating possible hashtags for a posted image automatically
- **Method**: Image Classification & Natural Language Processing

Demo

Separately uploaded on Github and Google Drive

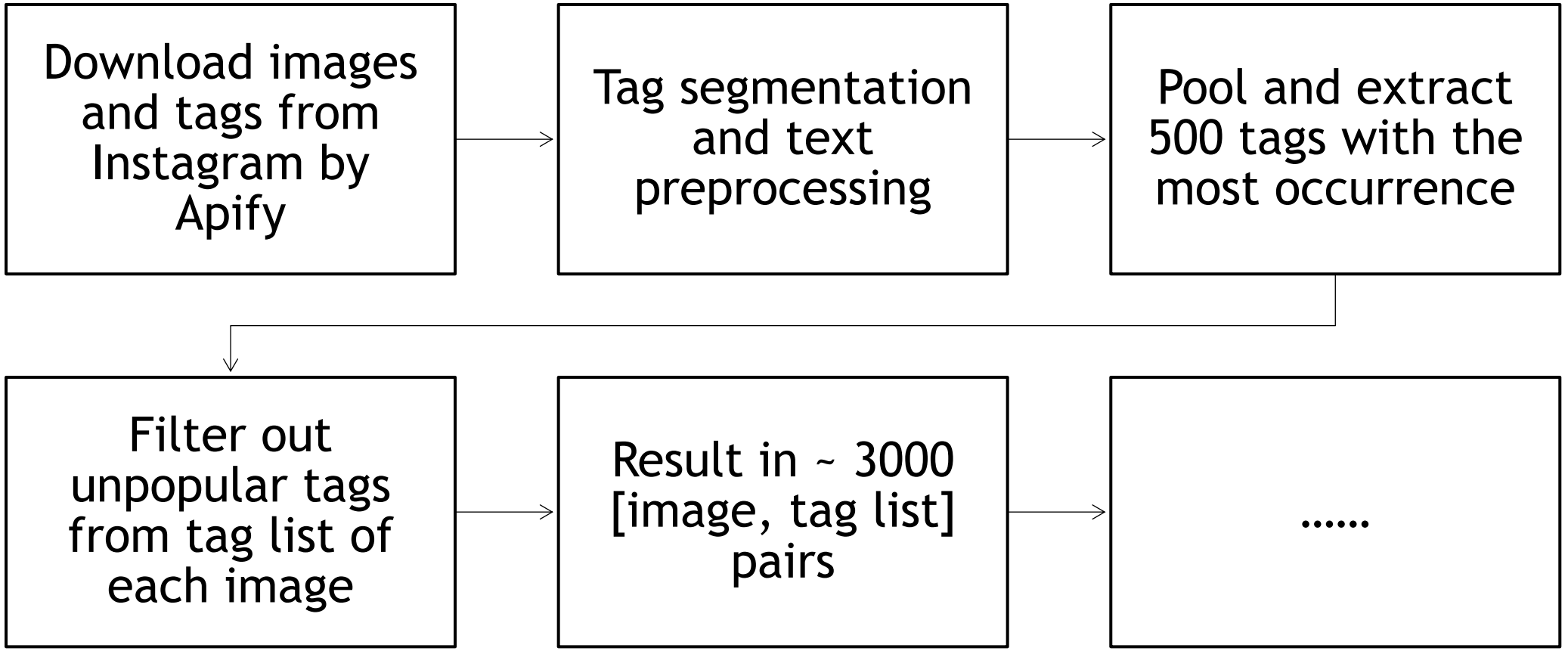
Data Source

- Hashtag
- Post Description
- First Comment
- Image URL
- Post URL
- Number of Likes
- Timestamp



alt	firstComment	imageUrl	likesCount	timestamp	url
Image may contain: dog and indoor	My little love bug ❤ #leothepuppy #mamasboy #bordercollies #doglover #rubmybelly	https://scontent-ams4-1.cdninstagram.com/v/t51.2885-15/e35/p1080x1080/83712001_864	0	2020-02-25T05:13:13.000Z	https://www.instagram.com/p/B8-kqwpHlLS

Data Preparation



Model: K-Dimensional Tree

Image classification

(Time: 0.02s/image)

- Apply pre-trained ResNet50 model (50 layer CNN trained on ImageNet) on 3000 image to get classification for each image

Sentence encoding

(Time: 0.001s/image)

- Encode the word classification of each image
- Encode the most popular 500 tags
- Construct a K-Dimensional Tree (a binary search tree in 512-dimension)

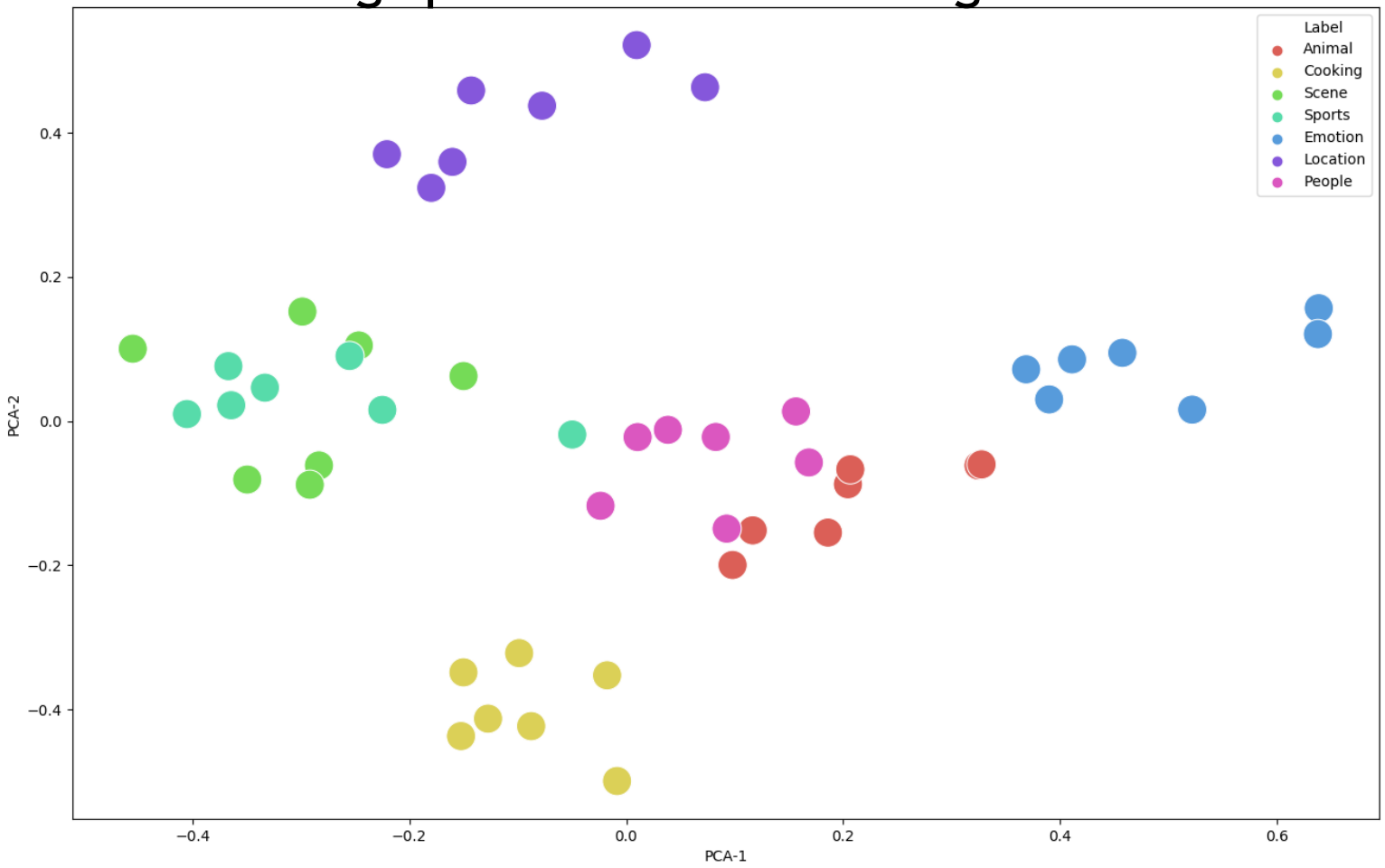
Nearest neighbors

(Time: 0.002s/image)

- Compute 2-norm (Euclidean) distance between 512-dimension vectors
- Locate the top 20 nearest neighbors in the tag pool for each image

Model: K-Dimensional Tree

Tag space with embedding + PCA



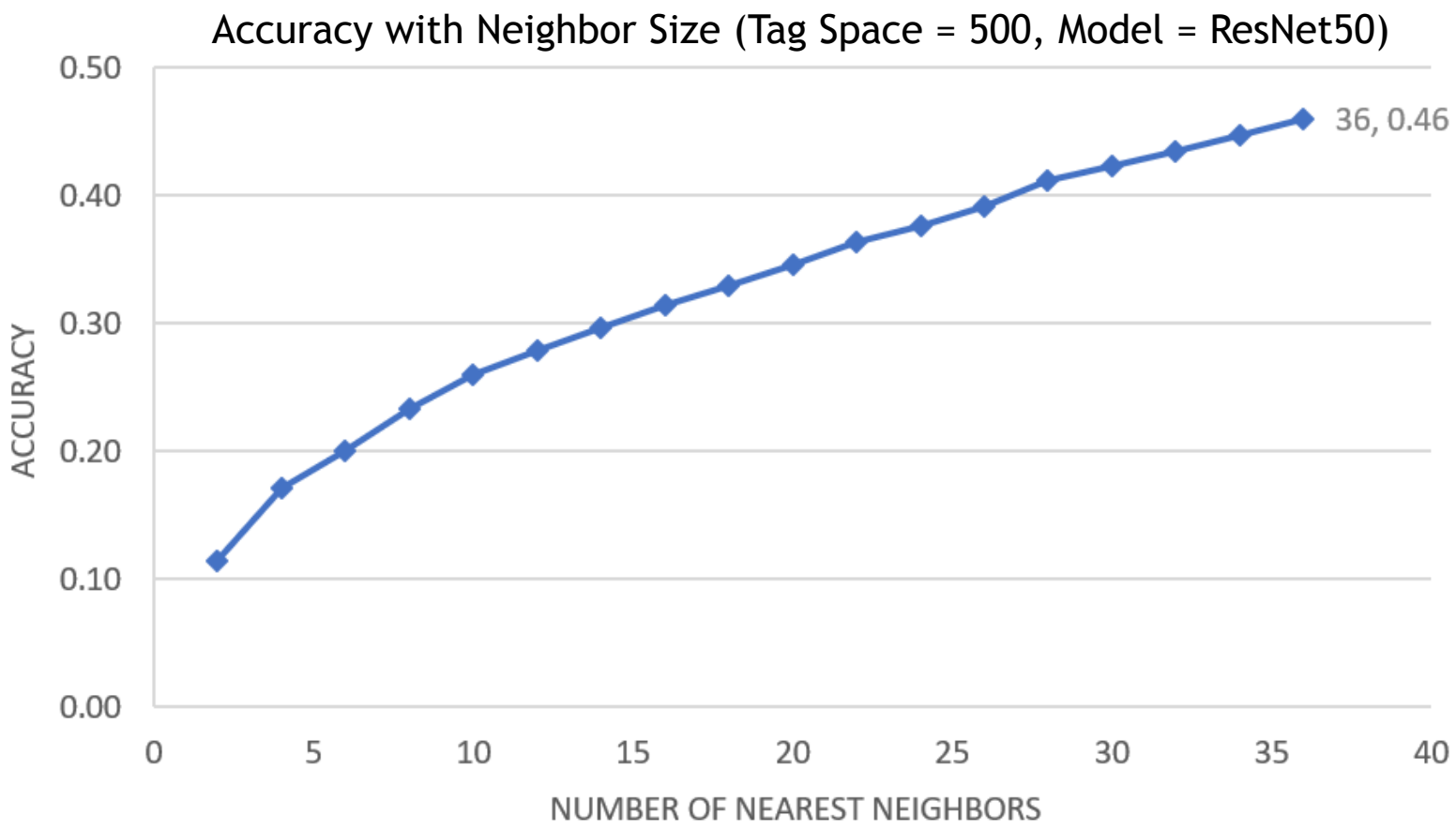
Model: K-Dimensional Tree

- ['#doglife', '#dog', '#puppylife', '#puppy', '#puppydog', '#puppygram', '#dogoftheday', '#doglovers', '#dogs', '#ilovemydog', '#dogstagram']

- ['#baking', '#cakedecorating', '#pastry', '#pastrychef', '#pastrylove', '#cake', '#cakeart', '#foodies', '#heels', '#cakedesign', '#breakfast', '#sushi']

Model: K-Dimensional Tree

- Accuracy: list of tag predictions overlap with list of true tags



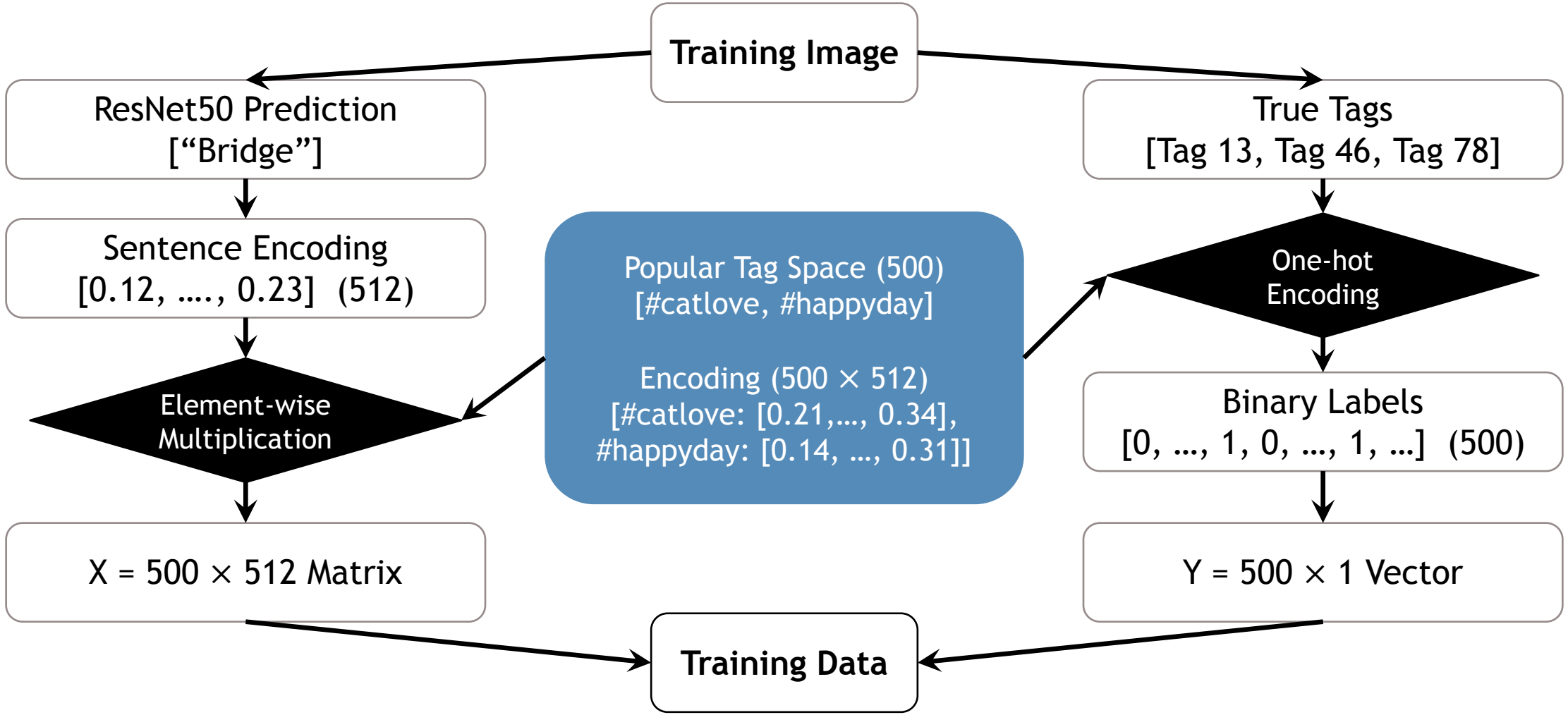
Model 2: Logistic Regression

$$\vec{a}, \vec{b} \in \mathbf{R}^k \qquad \|\vec{a}\|_2 = \left\|\vec{b}\right\|_2 = \|\vec{w}\|_2 = 1$$

$$D_{\text{Euclidean}}(\vec{a}, \vec{b}) = \sqrt{\sum_{i=1}^{k} (a_i - b_i)^2} = \sqrt{2 - \sum_{i=1}^{k} 2a_i b_i}$$

$$D_{\text{Weighted Euclidean}}(\vec{a}, \vec{b}) = \sqrt{\sum_{i=1}^{k} w_i (a_i - b_i)^2} = \sqrt{2 - \sum_{i=1}^{k} 2w_i a_i b_i}$$

Model 2: Logistic Regression



Model 2: Logistic Regression

Training	2300 Images
Down Sampling	Label 0 : Label 1 = 5 : 1
Data Matrix	X = 70000 × 512, Y = 70000
Runtime	6 Hours

Testing	600 Images
Classification Accuracy	0.9758
Accuracy per Image	0.0976
Runtime	1 Hour

What We Learned

- We were unclear about whether we should use classification or generative models. We discussed with professor and GSI and chose classification because it will perform well with the limited dataset we have.
- We tried several existed image tagging models available online, including VGG and ResNet, and we chose ResNet as the final model given its better performance on our dataset.
- On the teamwork wise, we learned how to split tasks based on team members strengths given the multiple aspects of our project: web scraping, natural language processing, image processing, and the UI design.

Thank you!

Questions & Comments?