Lab1
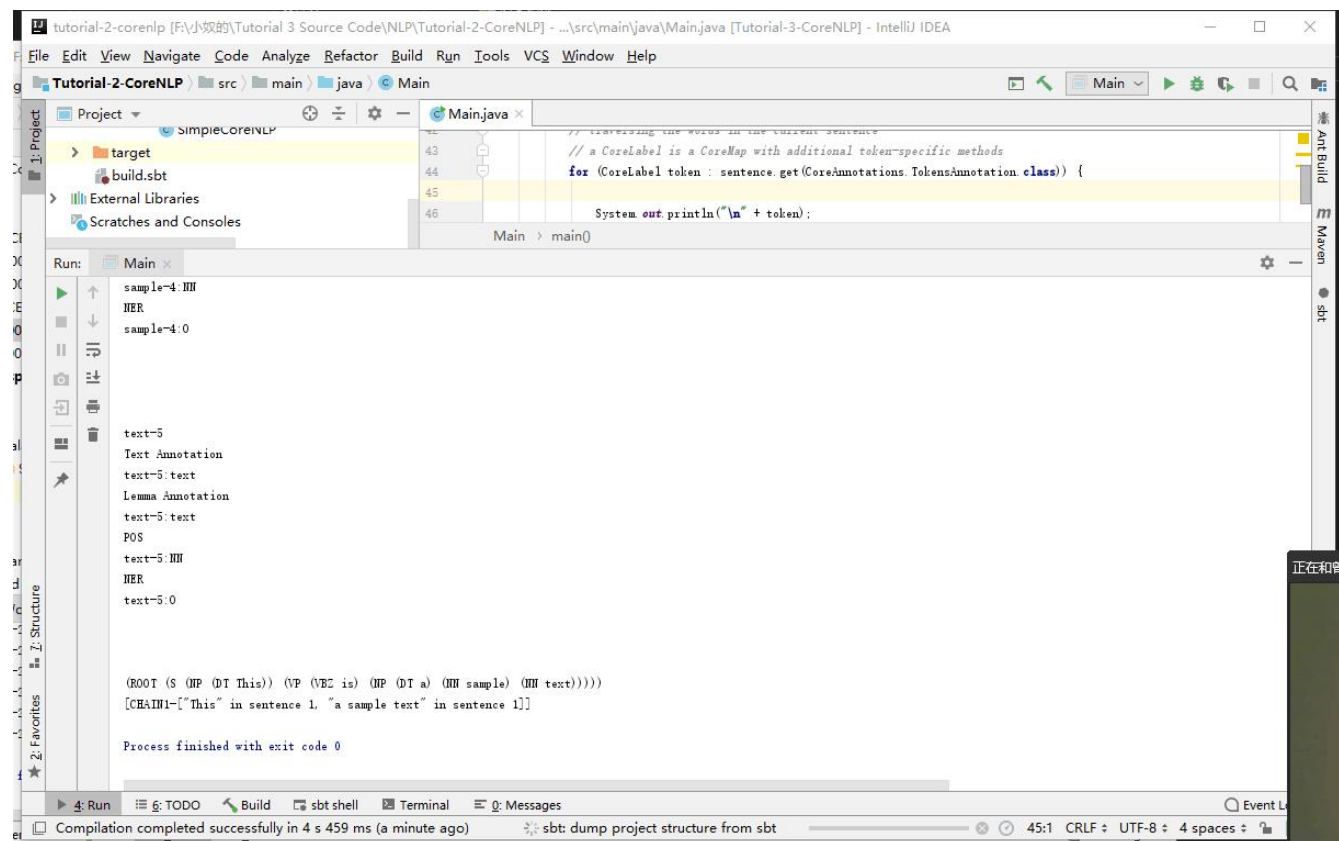
Yue Zeng

We use the Flickr 8k dataset and the animal theme for the natural language progress and SIFT.

1.NLP

2.SIFT Feature Extraction on cats images

CO Sift.ipynb ☆
File Edit View Insert Runtime Tools Help

COMMENT   SHARE

CODE   TEXT   CELL   CELL

CONNECT ▼   EDITING   ∧

```python
for m,n in matches:
    if m.distance < 0.75*n.distance:
        good.append([m])

# draw_params = dict(matchColor = (0,255,0),
#                    singlePointColor = (255,0,0),
#                    matchesMask = matchesMask,
#                    flags = 0)

# img3 = cv2.drawMatchesKnn(img1,kp1,img2,kp2,matches,None,**draw_params)

# cv2.drawMatchesKnn expects list of lists as matches.
#img3 = cv2.drawMatchesKnn(img1,kp1,img2,kp2,good,flags=2)
img3 = cv2.drawMatchesKnn(img1,kp1,img2,kp2,matches,None,flags=2)

plt.imshow(img3),plt.show()
```


(<matplotlib.image.AxesImage at 0x7f46de5a16a0>, None)

CO Sift.ipynb ☆
File Edit View Insert Runtime Tools Help

COMMENT   SHARE

CODE   TEXT   CELL   CELL

CONNECT ▼   EDITING   ∧

```python
matches = flann.knnMatch(des1,des2,k=2)

# Need to draw only good matches, so create a mask
matchesMask = [[0,0] for i in range(len(matches))]

# ratio test as per Lowe's paper
for i,(m,n) in enumerate(matches):
    if m.distance < 0.7*n.distance:
        matchesMask[i]=[1,0]

draw_params = dict(matchColor = (0,255,0),
                   singlePointColor = (255,0,0),
                   matchesMask = matchesMask,
                   flags = 0)

img3 = cv2.drawMatchesKnn(img1,kp1,img2,kp2,matches,None,**draw_params)

plt.imshow(img3,),plt.show()
```


(<matplotlib.image.AxesImage at 0x7f46da489898>, None)