	location	aspect	positive correct	positive total	negative correct	negative total	none correct	none total	positive percentage	negative percentage	none percentage	total percentage
0	LOCATION1	dining	30.0	30.0	1.0	2.0	1452.0	1458.0	100.00	50.00	99.59	99.53
1	LOCATION1	general	315.0	359.0	88.0	113.0	933.0	1018.0	87.74	77.88	91.65	89.66
2	LOCATION1	green-nature	33.0	40.0	0.0	0.0	1443.0	1450.0	82.50	NaN	99.52	99.06
3	LOCATION1	live	50.0	63.0	14.0	23.0	1373.0	1404.0	79.37	60.87	97.79	96.44
4	LOCATION1	multicultural	27.0	39.0	0.0	3.0	1445.0	1448.0	69.23	0.00	99.79	98.79
5	LOCATION1	nightlife	59.0	62.0	1.0	2.0	1411.0	1426.0	95.16	50.00	98.95	98.72
6	LOCATION1	price	67.0	81.0	112.0	116.0	1261.0	1293.0	82.72	96.55	97.53	96.64
7	LOCATION1	quiet	10.0	14.0	13.0	15.0	1456.0	1461.0	71.43	86.67	99.66	99.26
8	LOCATION1	safety	56.0	61.0	54.0	66.0	1340.0	1363.0	91.80	81.82	98.31	97.32
9	LOCATION1	shopping	57.0	62.0	1.0	1.0	1420.0	1427.0	91.94	100.00	99.51	99.19
10	LOCATION1	touristy	20.0	25.0	0.0	0.0	1454.0	1465.0	80.00	NaN	99.25	98.93
11	LOCATION1	transit-location	126.0	151.0	18.0	33.0	1272.0	1306.0	83.44	54.55	97.40	95.03
12	LOCATION2	dining	3.0	4.0	0.0	0.0	380.0	383.0	75.00	NaN	99.22	98.97
13	LOCATION2	general	71.0	87.0	20.0	26.0	245.0	274.0	81.61	76.92	89.42	86.82
14	LOCATION2	green-nature	5.0	7.0	0.0	0.0	376.0	380.0	71.43	NaN	98.95	98.45
15	LOCATION2	live	11.0	14.0	2.0	4.0	362.0	369.0	78.57	50.00	98.10	96.90
16	LOCATION2	multicultural	5.0	8.0	0.0	1.0	377.0	378.0	62.50	0.00	99.74	98.71
17	LOCATION2	nightlife	12.0	12.0	0.0	0.0	373.0	375.0	100.00	NaN	99.47	99.48
18	LOCATION2	price	21.0	27.0	26.0	27.0	328.0	333.0	77.78	96.30	98.50	96.90
19	LOCATION2	quiet	0.0	2.0	4.0	5.0	379.0	380.0	0.00	80.00	99.74	98.97
20	LOCATION2	safety	13.0	14.0	13.0	17.0	351.0	356.0	92.86	76.47	98.60	97.42
21	LOCATION2	shopping	10.0	15.0	0.0	0.0	371.0	372.0	66.67	NaN	99.73	98.45
22	LOCATION2	touristy	4.0	5.0	0.0	0.0	380.0	382.0	80.00	NaN	99.48	99.22
23	LOCATION2	transit-location	20.0	29.0	4.0	8.0	338.0	350.0	68.97	50.00	96.57	93.54

Figure 1: Results from BERT-pair QA-M model

Figure 1 shows the prediction results of the **BERT-pair QA-M** model on the SentiHood testing dataset. For every location-aspect pair:

- *positive total*, *negative total*, and *none total* represent the total number of opinions with positive/negative/none sentiments respectively in the testing set.
- *positive correct*, *negative correct*, and *none correct* represent the number of opinions that were correctly predicted by the model with positive/negative/none sentiments.
- *positive percentage*, *negative percentage*, and *none percentage* represent the percentage of opinions that were correctly predicted by the model.
- *total percentage* represents the percentage of correct opinions for each location-aspect pair.

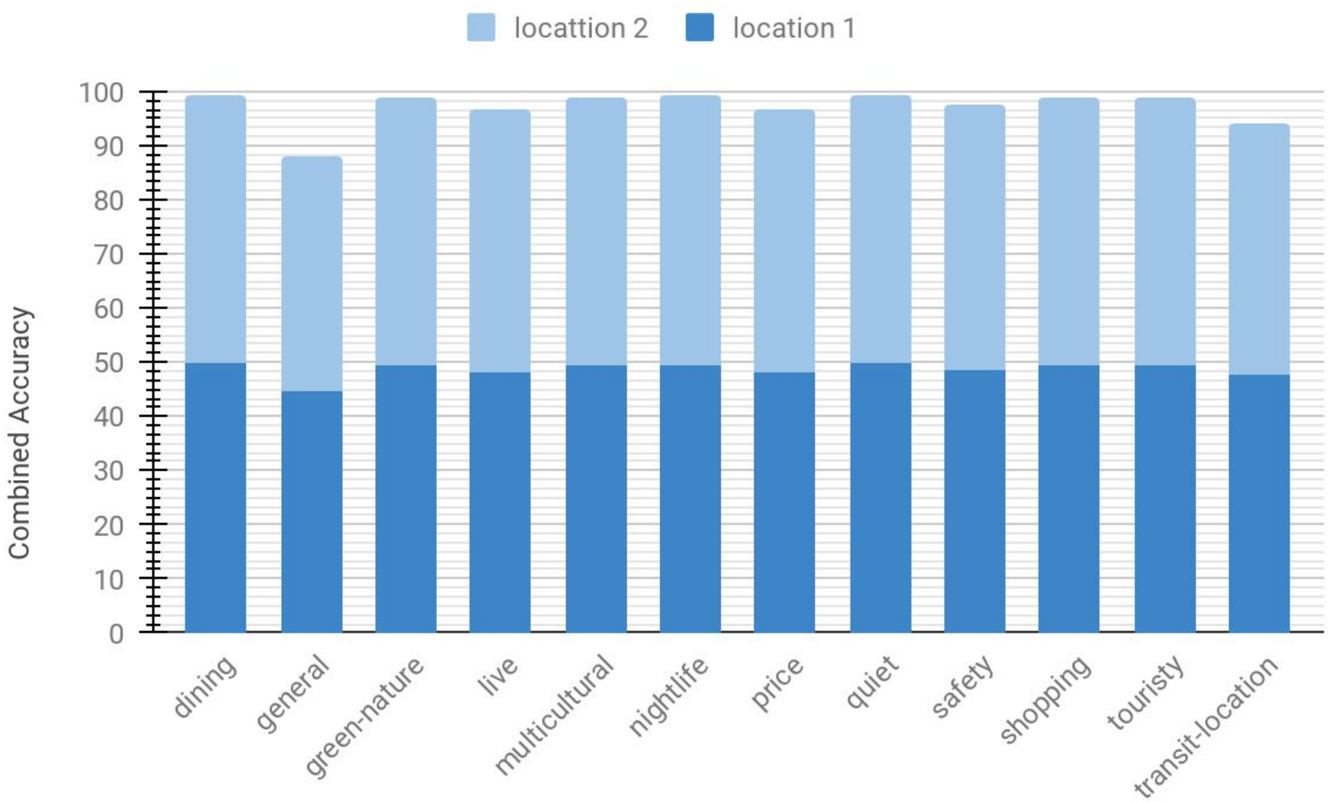

Figure 2: Combined accuracy of all aspects

➔ From figure 2, it can be inferred that the aspects *dining, nightlife,* and *quiet* get accurately detected by the model. *dining* has an accuracy of 99.53% for location-1 and 98.97% for location-2, making an overall of 99.25%, similarly, *nightlife* has an accuracy of 98.72% for location-1 and 99.48% for location-2 with an overall of 99.1% and *quiet* has an accuracy 99.26% for location-1 and 98.97% for location-2 with an overall of 99.12%. A common pattern in these aspects is the skewness among sentiments in the testing set.

➔ From figure 2, it can also be inferred that the aspect *general* is the bottleneck with a total percentage of 89.66% for location-1 and 86.82% for location-2. From figure 1, it can be said that the negative sentiment of aspect *general* has the worst performance. A relatively low *none percentage* suggests that the model often falsely detects this aspect in the text. A manual inspection of the testing set reveals that it is indeed relatively difficult to predict if a text is referring to the aspect *general* or not, as its definition is quite broad and subjective.

One of my favorite machine learning libraries is Scikit-Learn. Although it is the modus operandi for implementing machine learning algorithms, it does have a few limitations. Firstly, it lacks compatibility with many Deep Learning frameworks like PyTorch, e.g. I did not find any straight forward implementation to utilize the HuggingFace's transformers library with scikit-learn for this task. Secondly, it does not support GPU computation, which can reduce the training time for many learning algorithms. Lastly, it does not treat the training and testing steps differently, which restricts the use of techniques like dropout.