ML Intern Assignment

Problem Statement

Given a text and a *"*phrase*"* from it, detect the sentiment expressed towards the *"phrase"* in the text instance.

Examples
- cannot rely on both milk delivery and grocery | milk | Negative
- your customer service is terrible! | customer service | Negative
- I love notion as a tool | tool | Positive
- notion is a great site and an iPhone app. | notion | Positive
- Asked for a workspace name or billing email address | billing | Neutral

literature survey of research papers -

I didn't do that deep literature survey of papers for training procedures and other modeling inspirations but I have done some of my own intuition-based solution procedure. Though it should be necessary for doing the survey I did my best in solving the problem based on my thinkings.

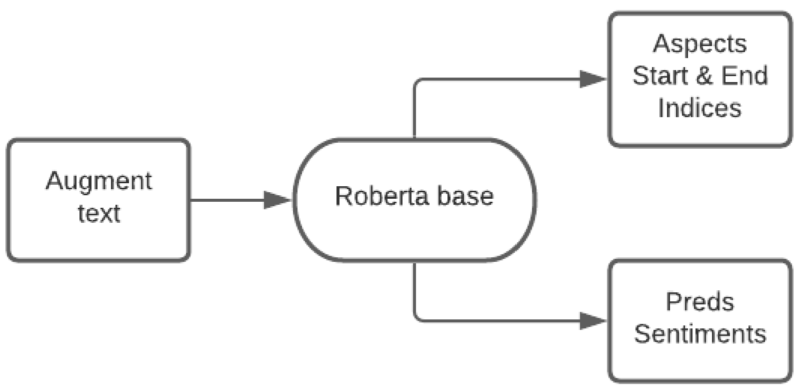
explain your approaches -
The first approach is a simple model where I have used a straightforward Roberta base model and the other part is of training a model that predicts the aspects and sentiment at the same time.

1. Straight forward - Here I have trained Roberta base model with cross-entropy loss with a basic text augmentation
   Eg - **"cannot rely on both milk delivery and grocery"** is converted to
   "cannot rely on both milk delivery and grocery. Aspect: milk"
   What's the advantage of this it increased the accuracy score by **approx 3-4%** with respect to the training on simple texts provided also decreases the over fittings.

   To describe with more detail the procedure defined is similar to any text classification problem where I have experimented with original text and Slightly augmented ones. this dataset is then split by **stratified k-fold where k = 10.** The model architecture is defined in a linear fashion.

Here the augmented text is the text where I have added the aspects explicitly and then this text is then fed to the model with getting sentiments prediction

Multi-Task Learning - In this approach I have predicted both **aspect and sentiment from the text** so the procedure of training becomes different. The model used is the same. The procedure resembles a **QA** system where the text is a question and the Aspect is the answer. With another head producing sentiment as output also. So this model produces both starting and ending indices and sentiments.
So basically I have created a model producing both of the **aspects and the respected sentiments**
Model architecture



To define the model here i have created start and end indexes of aspects string present in the main text provided originally hence we got the indexes of start and end also the sentiment is there to predict.
This creates a 2 head architecture where the sentiment head is trained on cross-entropy and other head outputs start and end indices which are optimized by cross-entropy.

final metrics of each approach -

1. Accuracy score for 1st method
2. Jaccard score for aspects overlap, accuracy score for 2nd part

ablation study table
Model 1

accuracy (on val)	With augment text	Original text
Fold 1	.77644	.7455
Fold 2	.91106	.87981
Fold 3	.97356	.95192
Fold 4	.99038	.96635
Fold 5	.98804	.96875
Fold 6	.99063	.96394
Fold 7	.99032	.96875
Fold 8	.99279	.97115
Fold 9	.99760	.98077
Fold 10	.98037	.93750
Mean Score	**.96143**	**.93190**

Model 2

Validation eval	Jaccard score (aspect overlap)	Accuracy score (sent)
Fold 1	.4559	.7262
Fold 2	.4709	.7025
Fold 3	.4674	.7375
Fold 4	.4520	.7187
Fold 5	.4913	.7287
Mean Score	**.4675**	**.7227**

So if the 2nd model gives **a 47% Jaccard score** and to be a little more accurate than it's around **50%** of times that the provided target texts are in the selected text produced by outputs and the sentiment score is **72%** which is less than the first model but its multi-task solution so this could be a trade-off.

Error Analysis-
Model 1 though produces good results but the model still overfits we either could use small models like the Albert or distill Roberta and hence could really push the limits of the model.

While model 2's low performance on sentiment comes from the high loss from aspect predictions though this could be solved we have to do a good number of architecture changes like using CNN/lstm nets as heads for start and end index predictions. This is a good way to tackle the loss values from an aspect-based approach and hence leading to an increase in sentiment accuracy score and overall loss.

Conclusion-
The main contribution comes in creating a good and generic modeling procedure for sentiment predictions. Also, this approach is easy to deploy and produces **97%** accurate scores. The other model is multi-tasking which could be utilized efficiently and has a high application quotient. As this model not only produces the sentiment with **73%** and around **50% iou** for aspect predicted to actual at the same time though more research could be done on training procedure and architecture build of these models.