**TRAINING METHODOLOGY REPORT**

Submitted By - Hardik Arora

The task assigned to me was : given a “phrase” and “text” , I need to detect the sentiment

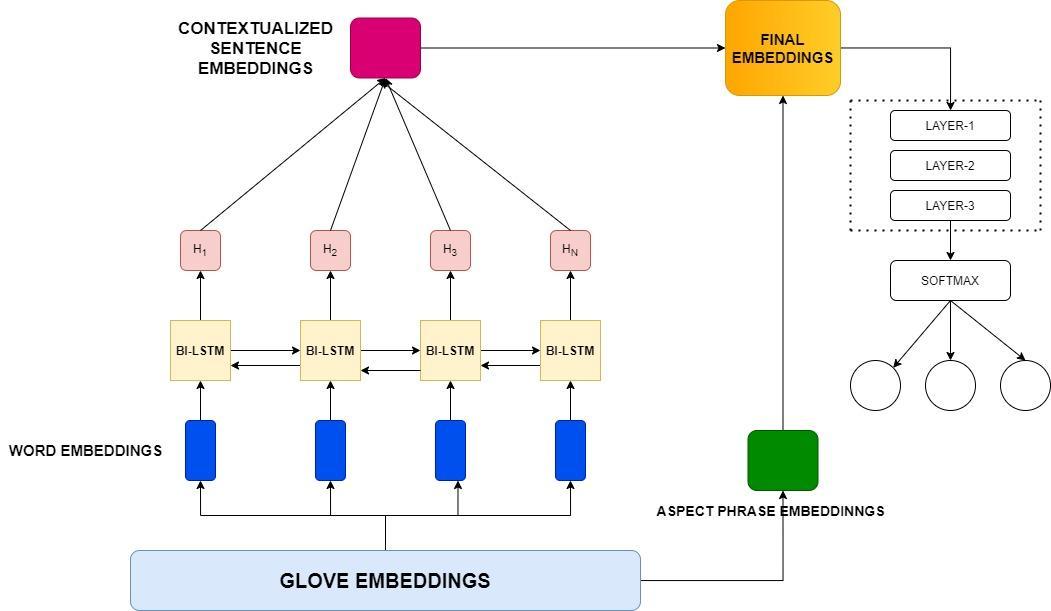
Expressed towards the phrase in the text instance.

**Data Analysis:**

A very basic analysis of the data can be found at the following html file.

**Training Approach**

For this task,I propose a deep learning based approach. The model architecture is shown below:



MODEL DESCRIPTION

* **Word embeddings layer** - For calculating embeddings for the sentence as well as for the aspect phrase I tried different ways :

1. Transformer based embeddings(Bert and Roberta)
2. Normal nn.Embeddings layer available in pytorch without any externa weights
3. Static Glove Embeddings **(finally used in model)**

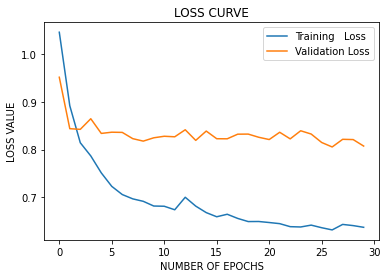
The problem with **approach 1** is that it gives you an embedding with 768 features ,which is practically possible for the model to learn due to less training data

The problem with **approach 2** was better than approach 1 but was not giving good results maybe again due to lack of learning

Due to all this I used **approach 3** for creating the word embeddings , I used standard 6B.100d glove version for creating 100 dimension ,sentence and word embeddings , I also tried with 50 dimension and 200 dimension but this was giving the best results in our favour.

* **BILSTM LAYER-** Since I used static word embeddings , words are actually deprived of any context in their vicinity. To solve this I used a simple bi-lstm network for generating contextualized word embeddings
* **Feed Forward layers-**  After getting embeddings for both the aspect phrase as well as for the text , I used some linear layers so that the model could find useful features and establish relationships between input and output.
* **Softmax Layer-** At last I used a softmax layer for predicting the sentiment.

Since this is a multi class problem I used a cross entropy loss as my loss function . I used Adam optimizer with a learning rate of 0.001 . All the dropout layers have a dropout rate of 0.3. I trained the model for 30 epochs ,with a batch size of 16 on Google Colab.

****

**Evaluation Metrics:**

For the purpose of evaluating my model I used f1 score (both label wise and overall)

Here are the scores for my model on the validation set:

|  |  |  |  |
| --- | --- | --- | --- |
| Models | Precision (weighted) | Recall (weighted) | F1-Score |
| Model-Glove 50 | 0.70 | 0.70 | 0.70 |
| Model-Glove 100 | 0.72 | 0.71 | 0.71 |
| Model-Glove 200 | 0.72 | 0.72 | 0.72 |

**Class wise results**

FOR 100-DIMENSION GLOVE EMBEDDINGS

|  |  |  |  |
| --- | --- | --- | --- |
| Class | precision | recall | f1-score |
| 0 | 0.79 | 0.67 | 0.73 |
| 1 | 0.62 | 0.78 | 0.69 |
| 2 | 0.70 | 0.70 | 0.70 |

FOR 200-DIMENSION GLOVE EMBEDDINGS

|  |  |  |  |
| --- | --- | --- | --- |
| Class | precision | recall | f1-score |
| 0 | 0.78 | 0.73 | 0.75 |
| 1 | 0.65 | 0.78 | 0.71 |
| 2 | 0.71 | 0.62 | 0.66 |

I preferred using 100 dimension embeddings due its more f1 score on the positive class ,and as mentioned the score for positive class has more importance.

**Error Analysis**

1. **Confusion due to neutral looking like negative words(or semantically less negative words)**

**Example-**

* This app works great but could you please PLEASE add an audio alert?

Aspect Name- audio

Predicted- neutral

True-Negative

* I wasted so much time with different project managers itâc™s ridiculous because the air table was right there all along.

Aspect Name- project

Predicted- neutral

True-Negative

* Less in the free version, it has enough features to work on projects.

Aspect Name- free

Predicted- neutral

True-Negative

**2) Aspect term itself has some negative sentiment**

* good afternoon, i would like to remove a logo i no longer use, its in the logo section.

Aspect Phrase- remove

Predicted- Negative

True- Neutral

**3) Some errors in the dataset itself**

* notion for desktop, web and mobile is good, but the ipad version is not good, interface is not properly scaled for the screen size and resolution.

Aspect Term- Ipad version

True - Positive

Predicted- Negative

* google docs and sheets have this.

Aspect Name- Sheets

True- Negative

Predicted- Neutral

* Love this app and it works supertly for planning work.

Aspect Name- planning

True-Neutral

Predicted-Positive

**4) Some vague sentences**

* one of the main principles of gtd is having all of the todo list in one place seems simple but has so much benefit.

Aspect Name- todo list

True- Positive

Predicted- Negative

**5) Biased a little towards negative words**

* great tool overall, but booting up the application on android takes way too long.

Aspect Name- tool

True - positive

Predicted - negative

* Can't we add reminders to tasks in the free version?

Aspect Name- free

True - neutral

Predicted - negative

* i want to clear my card details for my profile, can you help me on here?

Aspect Name- profile

True- Negative

Predicted- Neutral

**Conclusion:**

I just want to conclude that this problem statement is a data dependent problem statement.

Having given more data the results would have been a bit better.

**Future Ideas**

1. We could make use of dependency parsing trees ,generated using spacy to extract direct relationships between the aspect phrase and other words in the sentence.
2. We could use attention for making the contextualized embeddings even better.
3. If given more data we can even think of creating a dictionary of positive and negative sentiment words , which might be of some help ,not sure.