MAIS 202 – Project Deliverable 3

Final Training Results

Final Training Results:

I also tried to implement an RNN (LSTM). For that, I enumerated all possible words in the vocabulary by the ascending order of their count. That meant that every word in a review was assigned a number. Then, I transformed all reviews to lists of numbers. If a word had a number of higher than 20,000, I replaced it with number 0 to limit the size of the vocabulary. After that, I padded all reviews to a length of 50 words. So, if a review was less than 50 words, I would fill all the missing spots by 0 to make it reach a size of 50. If a review had more words, then any word that exceeds pos 50 was cut out. The reason I choose 50 was because 75% of the reviews had 21 words or less, so I choose 50 just to be safe . I then took a sample of 100,000 reviews. 80% were used for training the model. The model had an embedding layer with a 100 output length, LSTM with 128 nodes and a final dense layer of 3 nodes. I tried different values for everything and multiple dense layers. No matter what I did, the accuracy was stuck at around 68%. For that reason, I decided to go back to using SVM as it had 88% accuracy.

Finally for the aspect based analysis, the data didn't come labelled with aspects. Thus, there was no learning involved in this step. What I did was use nltk to extract the subjects of the sentences, then import the sentences they were included in to svm.predict to get the predictions on each aspect. However, there is no way for me to measure the accuracy of this algorithm and so I feel like I should no longer show the aspect based part in my analysis as it might greatly affect the accuracy of the results.

Final demonstration proposal :

I tried different things for the web app and it was a rollercoaster of an experience.

First, I tried Django. Django is really tedious and it involves many steps that are just too much for a simple web landing page. I spent hours upon hours trying to figure things out, however their web forms process isn't clear and you can easily get lost if you don't have experience. I learned lots of stuff about Django, mainly that I don't want to ever see it again.

Second, I tried using CGI under my given cs domain. That was because the process was familiar from comp 206, even though I learned that it was a bit outdated. There were errors, and figuring them out was only a matter of guessing so I just left it at that. You don't know how the server works so it's really time consuming to know what to do.

Finally, I used flask. This channel was a God send:
https://www.youtube.com/channel/UCN7zakbo7GQBT5RkQWhpm4A/playlists. After spending hours upon hours on CGI and Django, I was able to finish my work in under an hour because of his instructions. Flask is simple. I love flask and I despise Django. Here, I saved my svm models and vocab. Homepage asks the visitor for a review, they type it down and then it tells them whether they like their phone or not. As simple as.