Contest 1 Experiment Report

อรรถพล ธำรงรัตนฤทธิ์

Experiment Setup

The dataset contains XXX rows. The text is drawn from restaurant reviews. We clean the dataset by looking at XXX and filtering out the reviews that are XXX. We exclude these rows because they XXX. And they might not fit into the model. We also clean out stopwords, numbers, and other non-English alphabet symbols because we think that they are not relevant to the task and might introduce noise to the system.

We split the dataset into XX:XX:XX proportion. The training set has XXX rows. The development set has XXX rows. And the test set has XXX rows. The label distribution for the three sets is as follows:

	training set	dev set	test set
positive			
neutral			
negative			
conflicting			

We observe that XXX label does not have enough samples for proper model training. So we formulate this task as a X-way classification. We only classify text into positive, neutral, and negative.

As far as aspect classification, we simplify the task into two-way classification. A text is classified into either food or service to make this task simpler albeit sacrificing the accuracy in the other aspects in the test set. So the overall dataset size is smaller than the sentiment analysis task where no rows are excluded.

	training set	dev set	test set
food			
service			
XXX			

XXX			

Model

As for baseline models, we chose a bag-of-word logistic regression model for both sentiment and aspect classification. The features are created from the cleaned text as described in the previous section. We also use bigram features as we think that bigrams might be able to capture more multiword complex expression of sentiment.

We hypothesized that deep learning models can achieve higher accuracy due to its use of pretrained word embeddings. We experiment with Google News 300-dimensional word embeddings, which are freely available. We also compare the performance with GloVe word embeddings. We use Deep Averaging Network (DAN) and Convolutional Neural Network (CNN) for both tasks.

Hyperparameters are tuned in a grid search. We use AdaGrad optimizer and set the learning rates to 0.001, 0.005, 0.0001. The minibatch sizes are 64, 128, and 256. For CNN, we try many four filter sizes (2,3,4, and 5) at 100, 200, and 300 filters. For DAN, we use Dropout with p = 0.5, 0.6, and 0.7 to prevent overfitting at the hidden layer level. The word embeddings are allowed to be fine-tuned during the training process.

Results

The sentiment analysis task is far more difficult than the aspect classification task. The overall sentiment F1 is XXX, and the overall aspect F1 is XXX across all of the models.

Deep learning models outperform MaxEnt bag-of-word and bigram models regardless of the choice of embeddings. In fact, both Google and GloVe embeddings produce very similar results for both tasks and for both models.

Overall, CNN works better than DAN for sentiment analysis task. CNN overall sentiment F1 is XXX, and DAN oversall sentiment F1 is XXX. However, the aspect classification task does not show a significant advantage of CNN over DAN. This might be because we simplify the aspect classification into two-way classification. For these two labels, it might be sufficient to use a simpler model where word ordering does not affect the classification.

	positive F1	**negative F1**	**neutral F1**	**overall sentiment F1**
MaxEnt bag-of-word				
MaxEnt bigram				
DAN + Google				
DAN + GloVe				
CNN + Google				
CNN + GloVe				

	food F1	**service F1**	**overall aspect F1**
MaxEnt bag-of-word			
MaxEnt bigram			
DAN + Google			
DAN + GloVe			
CNN + Google			
CNN + GloVe			

Conclusion

From our experiment results, we find that CNN with Google embeddings performs the best for both sentiment analysis and (two-way) aspect classification tasks. Therefore, we use this model to classify the test set and submit to Gradescope. Our group achieves the F1 scores of XX and XX for sentiment analysis and aspect classification tasks respectively. We are ranked 4th on the leaderboard.