SI 630 Project Report

Shaoze Yang
shaozey@umich.edu

1 Introduction

As electric vehicles (EVs) continue to gain popularity, the importance of charging stations has become increasingly paramount. However, when it comes to selecting the best charging station, individuals often find themselves relying on popular mapping applications that provide only an overall score and numerous user comments. Sifting through these comments to determine the most suitable charging station can be both time-consuming and demanding. To address this challenge, I propose to develop a multi-class classifier based on the user comments associated with charging stations to perform a quantitative analysis. The goal is to identify specific aspects that users may find appealing or unappealing, such as charging speed, cost, and the quality of infrastructure at a particular charging station. Ultimately, this approach will enable individuals to make efficient choices by simply reviewing these categorized insights.

2 NLP Task Definition

This project aims to develop a classifier that maps comments to a predefined set of labels, encapsulating various aspects of charging stations. These aspects include accessibility & availability, amenities & location, compatibility & connectivity, among 16 distinct categories. Additionally, the system is designed to generate a corpus for each label. I may integrate the extracted keywords to a search engine for users to efficiently search for charging stations.

To illustrate how this system operates, consider the following example:

Input: Had trouble connecting initially. I called customer service. They rebooted the machine and it worked just fine after that. It is a paid service. Close to the Papa Murphys Pizza store. Uncovered. Would be accessible at any time day or night. Well lit parking lot. This is one of the better charging stations. Seems to be cheaper sometimes n less crowded, maybe? They're all pretty crowded.

Outputs (Aspect / (Sentiment) / Detected phrases):

1. compatibility and connectivity / Negative / Had trouble connecting initially
2. customer service / Positive / They rebooted the machine and it worked just fine after that
3. payment Options / Positive / It is a paid service
4. amenities and location / Positive / Close to the Papa Murphys Pizza store, walking distance to Target
5. accessibility and availability / Positive / Would be accessible at any time day or night
6. ...

Extracted Keywords: Compatibility issues, Customer service, Payment, Amenities, Location, Accessibility, Lighting, Pricing, Crowding

3 Data

The data comes from reviews about the stations in the former dataset from Google Maps. It would contain fields such as name, business status, and most important, detailed review content. There is an API provided by SerpApi that helps us to grab data. The data would come in the form of text documents. The collection contains 16767 records in total. The basic statistics of the data are illustrated in Figure 1.

The ChatGPT API automatically generates the training and validation sets, and I perform manual filtering in two stages. However, due to API speed constraints, I have managed to generate data for 6,000 charging stations and 38,484 sentences. An example provided by GPT is discussed in Section

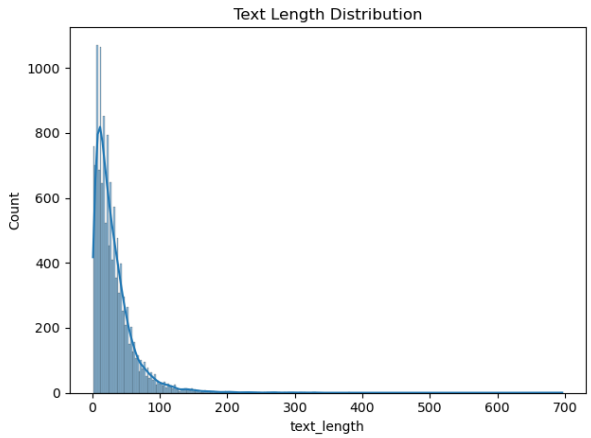

Figure 1: Distribution of Comment Length

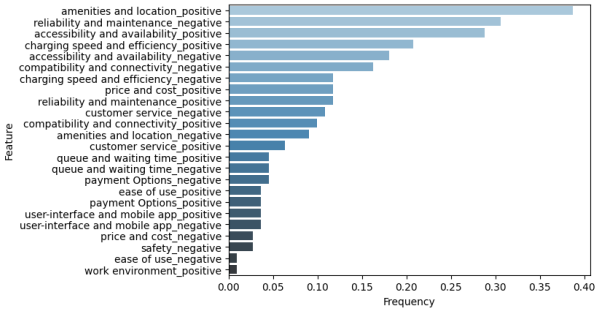

Figure 2: Counts of Extracted Aspects

2. **My goal is to develop a model that is significantly lighter than GPT yet maintains comparable accuracy.** Statistics for the extracted aspects are depicted in Figure 2.

4 Related Work

I am particularly interested in leveraging the attention mechanism, as it aligns with my objectives. Through observation, I've noted that GPT can pinpoint aspects from just a sentence or even a phrase within a lengthy comment. This is a significant advancement over previous models that relied solely on keywords. The attention mechanism, as outlined by Vaswani et al. in their seminal paper [3], allows for more resources to be concentrated on relevant areas to extract detailed information about the target while minimizing the impact of irrelevant data. Furthermore, Yang et al. introduced an attention-based document classifier [4], adding to the body of research supporting this approach. I've also explored various methods, including LSTM-based RNN classifiers [5] and CNN-based techniques. Should time allow, I plan to incorporate these methodologies into my project.

5 Methods

In this project, I aim to reproduce the methods in [4], which includes an hierarchical structure of attention. To achieve this, I would firstly implement a data loader, which could efficiently divide each review (as a document) to sentences. I will then implement a clipper which discard too long sentences or document by setting the threshold for maximum length.

5.1 Word Embedding

For the embedding of each word, I will use the gensim pre-trained model ([2]) and then train it through my collection of data. As a result, each embedding has the dimension of 300.

5.2 GRU-based sequence encoder

According to Bahdanau et al. [1], a GRU-based encoder would be implemented in a token level to efficiently track the state of sequences without using separate memory cells. In pytorch, this could be easily added by *torch.nn.GRU*. As a result, we are able to get the bi-directional summarized contextual information in the annotation for each token. The forwarding hidden state and backward hidden state are concatenated as completed information about a word.

5.3 Hierarchical Attention structure

The attention mechanism enable us to add weight to each token/sentences. I will discuss the token level attention here for simplicity:

$$u_{it} = \tanh\left(W_w h_{it} + b_w\right)$$

$$\alpha_{it} = \frac{\exp\left(u_{it}^\top u_w\right)}{\sum_t \exp\left(u_{it}^\top u_w\right)}$$

$$s_i = \sum_t \alpha_{it} h_{it}.$$

The word annotation from GRU will be further concatenated with the word embedding. The vector will pass a MLP to get the final hidden representation u_i. The attention would further generate a sequence of weighted representation. I will regard the sentence as the mean pooled result of all tokens inside. The GRU and sentence-level attention would be further implemented to get the representation of a document. Finally, a linear layer would map the document embedding to output labels.

5.4 Customized Data Loader

As mentioned, to incorporate a hierarchical attention model, we need to firstly separate the document (a comment) to sentence then tokenize the sentence into words. In this project, nltk sentence tokenizer and word tokenzier are used. We use the third quartile as the maximum sentence length. For sentence longer than the threshold, we clip the rest; for those shorter than the threshold, we use the placeholder for blank area.

6 Evaluation and Results

The scenario where each instance can belong to multiple classes with varying degrees of membership (e.g., counts or intensities rather than binary indicators), this moves beyond the standard multi-label classification framework. I have developed 2 methods to evaluate the performance of my classifier.

Firstly, I would compare the result of the generated labels with the ground truth. Since we have 16 different aspects, then we can describe a comment in a vector in 16 dimensions:

$$C = [A_1, A_2, ..., A_{16}]$$

We can calculate the distance of this vector to the vector of ground truth in L2 norm. Notably, this may also be used as loss function. Moreover, I can generate the confusion matrix of this result to further analyze which 2 or aspects are hard for model to distinguish.

Secondly, I will use a metric similar to the "hamming loss". Moreover, I made following adjustments:

1. Normalize the Predictions and Labels and ignore negative value in the predictions

$$y_i = \frac{clip(y_i, 0, max(\mathbf{y}))}{max(\mathbf{y})}$$

2. Adding threshold to convert the prediction ground truth to binary values

$$y_i = \begin{cases} 0 & y_i \leq threshold \\ 1 & y_i > threshold \end{cases}$$

In this way, I could not only evaluate how good the model predict the desired aspects through hamming loss, but also how good the model match the quantitative value in each aspect through L_2 Loss.

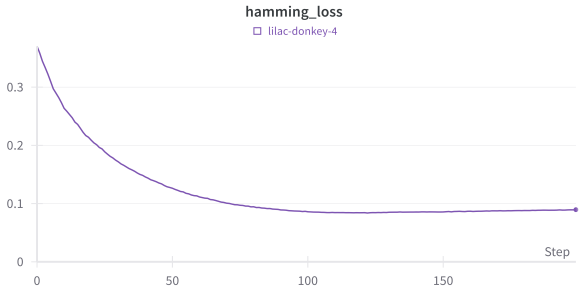

Figure 3: Modified Hamming Loss for Baseline Model

6.1 Baseline Model

I opt to use 3 baselines in this project. Firstly, I can test the accuracy of just choosing the most common label, the amenities and location. Also, since I would probably encode the comment to a vector, then I can also use other basic classifiers in scikit-learn such as decision tree, lightgbm, neural networks to see the corresponding performances.

At the same time, I would try the attention model similar to that we developed in howework2 as another baseline model. I also make several modifications for the new task. Firstly, I changed the loss function from BCELoss to Cross Entropy Loss to cope with multi-class classifcation task. Moreover, I have implemented the loss function as mentioned in previous section. I run it for 200 epoches and record the performance on development set every epoch, as shown in the Figure 3.

6.2 Hierarchical Attention Model

For this model, I use grid search for several parameters and record the model performance on validation set and the train loss for 2-hours training. Here are several parameters I chose:

1. Learning rate: 0.0001, 0.001, 0.01(origin), 0.1
2. Batch size: 2, 4, 8, 16, 64, 128(origin)
3. Optimizer: SGD(origin), AdamW

Among these, I found one combination of parameter outperforms, where lr equals 0.01, batch size equals 4, AdamW as optimizer. Here are the training loss, as shown in the Fig. 6; as well as the test loss, as shown in Fig. 4.

During the experiment, I found that the batch size greater than 8 fails to converge. As shown in the Fig. 7, the training loss has no significant change as training epochs increasing. Moreover, the model accuracy on validation set has increased after short training.

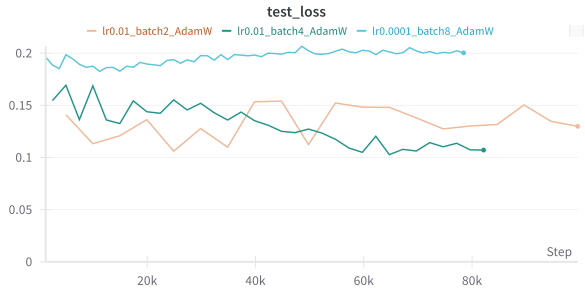

Figure 4: Modified Hamming Loss for Hierarchical Attention Model on the Test set.

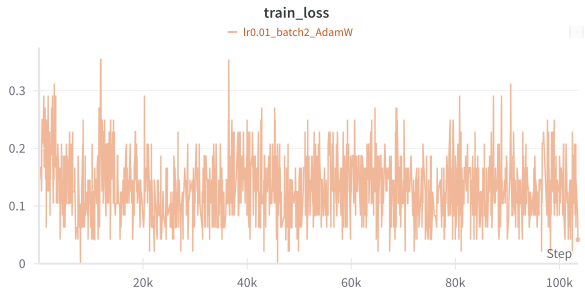

Figure 5: Modified Hamming Loss for Hierarchical Attention Model with batch size 2 on the Test set.

Figure 6: Modified Hamming Loss for Hierarchical Attention Model with batch size 4 on the Training set.

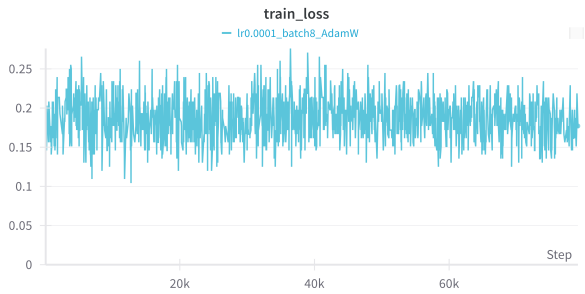

Figure 7: Modified Hamming Loss for Hierarchical Attention Model with batch size 8 on the Training set.

7 Discussion

During the experiment, several interesting observations were noted. Notably, as the batch size increased, model convergence was hindered. This phenomenon may be attributed to a variety of factors. Primarily, the variable length of comments implies that training data within each batch largely consists of either placeholders or truncated comments. Additionally, the intrinsic quality of comments, which typically do not match the coherence and structure of standard documents, poses challenges for effective parsing and utilization. Consequently, only a subset of high-quality comments contributes positively to model training. A larger batch size exacerbates this issue, as the impact of lower-quality comments disproportionately affects overall model performance.

8 Conclusion

The development of a hierarchical attention structure for the task of comment classification has yielded promising results. Evaluation on the validation set indicates that this model performs satisfactorily, surpassing the established baseline. Furthermore, several evaluation metrics were developed and tested, allowing the use of a comparatively lighter model with reduced computational requirements to achieve results on par with those of GPT-3.5. These classified results will be integrated into a recommendation system to provide personalized suggestions for charging station candidates tailored to different users.

9 Other Approaches Explored

Throughout the model development process, several alternative approaches were explored. Initial trials involved using models from the Hugging Face library, similar to those we experimented with in Homework 3. However, these models produced poorly formatted results, complicating the extraction of specific aspects and their corresponding scores. Additionally, their performance did not meet the standards of the baseline model, leading to the decision to discontinue their further development.

10 What you would have done differently

If once again, I would try use other GRU blocks such as LiGRU *lightGatedRecurrentUnits* or LiBRU *lightBayesianrecurrentunit*, whose reported

accuracy are higher compared to the GRU as I used in the model especially for speech classification tasks.

11 Future Work

The training process revealed considerable instability in train loss, which may stem from fluctuations in comment quality and inaccuracies in sentence tokenization. Consequently, it may be sufficient to utilize token-wise attention alone for simpler structures such as comments. In future work, I plan to simplify the model to a single-layer attention mechanism to evaluate its effectiveness. Furthermore, efforts will be made to develop a personalized recommendation system based on the classified results, enhancing the user experience by tailoring suggestions more closely to individual preferences.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473* (2014).

[2] Radim Rehurek and Petr Sojka. "Gensim–python framework for vector space modelling". In: *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic* 3.2 (2011).

[3] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[4] Zichao Yang et al. "Hierarchical Attention Networks for Document Classification". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kevin Knight, Ani Nenkova, and Owen Rambow. San Diego, California: Association for Computational Linguistics, June 2016, pp. 1480–1489. DOI: 10.18653/v1/N16-1174. URL: https://aclanthology.org/N16-1174.

[5] Chunting Zhou et al. *A C-LSTM Neural Network for Text Classification*. 2015. arXiv: 1511.08630 [cs.CL].