Dimensional Aspect-Based Sentiment Analysis (DimABSA)

Glen Qin, Nathan Khazam, and Yuri Fung
University of Colorado Boulder
{glqi1039, nakh6169, yufu7119}@colorado.edu
https://github.com/SUPERINTIALD/NLP_SEM_EVAL

Abstract

This paper describes our group's approaches, experiments, and findings for SemEval-2026 Task 3 Track A: Dimensional Aspect-Based Sentiment Analysis. Our best performing iteration used the XLM-RoBERTa, an improved version of BERT that utilizes the Transformer encoder architecture. By using various hyperparameter tuning methods, such as Grid Search and Bayesian Optimization, we found the best hyperparameters to be a learning rate of $2e^{-5}$, epoch count of 8, batch size of 16, weight decay of $5e^{-3}$, and dropout rate of 0.2. Through probabilistic analyses, we found that XLM-RoBERTa was our best model, and all models performed the best with higher resource languages such as English and Chinese. Given our testing, it's clear that each model needs its own set of tuned hyperparameters to achieve optimal results.

1 Introduction

Aspect-based sentiment analysis (ABSA) studies sentiment with respect to specific aspects or attributes mentioned in text (reviews of various industries). Traditional ABSA framed sentiment with discrete polarity, either positive, neutral, or negative (Peng et al., 2020). DimABSA aims for a multidimensional context and frames the problem in a continuous, dimensional space (variance, arousal), requiring finer-grained modeling and evaluation across multiple domains. This new formulation reflects psychological models of affect and provides richer signals for downstream tasks such as customer analytics and product triage (Buechel and Hahn, 2017). In this study, we examine several Transformer-based models such as XLM-RoBERTa and mDeBERTa. Based on our experimental results, we chose XLM-RoBERTa as our base model and tuned its various hyperparameters for English, Russian, Japanese, Ukrainian, Chinese, and Tatar. We employed Grid Search and Bayesian Optimization for hyperparameter tuning.

2 Background

As mentioned above, ABSA is used to identify sentiment about specific aspects or attributes of text. This idea initially arose from earlier SemEval tasks, such as those in 2014 - 2016, focused on aspect extraction and sentiment classification (Pontiki et al., 2015). Earlier methods often used SVMs or human-made features, which do not perform as well as modern ABSA systems. Using BERT-style models, DimABSA is able to create and evaluate multiple interpretable dimensions to classify text against. This allows for a more detailed sentiment analysis of text compared to older systems, which often used positive, negative, or neutral as their possible outputs (Mohammad and Bravo-Marquez, 2017). This modern approach allows for more information about sentiment, which can easily be related to multi-label sentiment classification, attribute-based evaluation (this task), or fine-grained classification. Modern approaches often use BERT-style models for evaluation since they perform quite well on this kind of natural language interpretation. As ABSA continues to improve, many new ideas and approaches have been developed, such as multi-output architectures for more than 2 dimensions and regression-based sentiment scoring. This task allows us to take a look at DimABSA development and provide our own ideas and interpretations for new versions of this technology.

3 System Description

3.1 System Overview

DimABSA focuses on using valence (degree of positivity) and arousal (intensity) scores to add dimensionality to standard ABSA. More specifically, the task requires determining Valence-Arousal (VA) scores for various industry reviews. Each VA is treated as a score along two dimensions: valence and arousal. The dataset includes annotated training data with IDs, aspects, and gold VA scores. The

evaluation metric used is the Root Mean Square Error ($RMSE_{VA}$) between the predicted and gold VA scores. Our system builds upon the following design decisions and components: leveraging pretrained multilingual models, fine-tuning to predict continuous VA scores, and hyperparameter tuning. More specifically, we focus on utilizing three BERT-based transformer architecture: RoBERTa, mDeBERTa, and BERT. Additionally, we utilized two hyperparameter tuning techniques: Grid Search and Bayesian Optimization.

3.2 Methods

Models. We use multilingual transformers XLM-RoBERTa and mDeBERTa. As a baseline, we employ BERT in order to compare the performance of these two models. By modifying the prediction head to output VA scores, these models provide cross-lingual embeddings suitable for multilingual VA prediction tasks.

XLM-RoBERTa. We utilized XLM-RoBERTa as its training pipeline removes the language-specific tokens used in other models. We also modified the output by adding a regression head with two outputs, valence and arousal. This regression head is a linear layer with a [CLS] representation.

$$\hat{y} = W^T(Dropout(h_{CLS})) + b \tag{1}$$

mDeBERTa. mDeBERTa extends the DeBERTa architecture through disentangled attention. By introducing this attention mechanism, this allows separation between content and positional embeddings, which improves contextual alignment and helps capture semantic differences.

$$A_{ij} = Q_i^C K_j^C + Q_i^C K_{i-j}^P + Q_i^P K_j^C \tag{2}$$

4 Experimental Setup

Dataset. To train our models, we used the original review data provided by the SemEval Task 3 organizers. This dataset consisted of English, Japanese, and Russian reviews of various industries (restaurant, laptop, hotel, and finance). To ensure generalization, we split the training data into [percentage] split of training and development sets for each language. This withheld data was used during hyperparameter tuning, to ensure that optimal search spaces were explored.

Grid Search. We begin with a structured grid search over a set of key hyperparameters. This provides a broad mapping of model performance and allows to identify viable regions for deeper search. We explore combinations of learning rate, epoch, batch size, weight decary, and dropout. We found the best initial hyperparameters to be a learning rate of $2e^{-5}$, epoch count of 8, batch size of 16, weight decay of $5e^{-3}$, and drop out rate of 0.2. Grid search was used to identify coarse trends, such as sensitivity to epoch count and variable learning rates to determine stability between languages.

Bayesian Optimization. Once promising regions were identified via grid search, we applied Bayesian optimization to efficiently explore high-performing mappings. We used the RMSE equation as our objective function to minimize with log-uniform and uniform search spaces for learning rate and dropout rates, respectively. This approach allowed us to focus on hyperparameter combinations that were predicted to yield better performance, limiting unnecessary training and providing more stable convergence than grid search alone.

Evaluation Metric. The performance of our system was evaluated on RMSE, which was provided by the task organizers. RMSE is calculated by comparing the predicted valence and arousal scores with the gold labels.

$$RMSE_{VA} = \sqrt{\sum_{i=1}^{N} \frac{(V_p^{(i)} - V_g^{(i)})^2 + (A_p^{(i)} - A_g^{(i)})^2}{N}} \tag{3}$$

Experimental Environments. We implement all our experiments using libraries such as PyTorch and HuggingFace. For pretrained models, we primarily use XLM-RoBERTa and RoBERTa. As discussed in the methods, we experiment with various hyperparameters using Grid Search and Bayesian Optimization.

5 Results

5.1 Hyperparameter Tuning

For our first set of results, we wanted to see how the hyperparameters affected the model. Initially, we manually tuned the system, using different learning rates and epoch cutoffs if we did not see a loss improvement after 3 more epochs. Our manual results were nothing special, and we noticed that the best learning rate centered around $2e^{-5}$. At

lower learning rates, the MSE increased by quite a bit, causing the model to perform noticeably worse. At higher rates, the model still learned quite well, but often had more outliers. $RMSE_{VA}$, the main metric of the project, hovered around 1.25 and was not a big help in picking a learning rate. This led us to choose $2e^{-5}$ for our initial learning rate.

Our next set of results was comparing RoBERTa vs mDeBERTa. We decided to run both models with a $2e^{-5}$ learning rate to see how they compared on the data. With mDeBERTa, there was much better clustering around the line of best fit compared to RoBERTa, leading to fewer outlier values. This led us to swap models and use mDeBERTa going forward with our tests on the provided training data.

5.2 Model Results

The final set of results focuses on the task at hand, where we trained and tested our model on multiple datasets. In total, we trained and tested on 7 of the provided data sets. However, we also decided to use automated hyperparameter tuning to improve the model's performance. Compared to our manually tuned hyperparameters, these models performed quite a bit better on the data, and we confirmed this using the English laptop dataset, which we had been using during our manual tuning. That also happens to be the first set of data we tested this system on.

Training dataset one was on English laptop reviews. With automated hyperparameter tuning, we achieved noticeably better results than we did with manual tuning. Our learning rate guess happened to be right, as the model used a $2e^{-5}$ learning rate. The dropout on the model was also quite low, at 0.05, and we had an $RMSE_{VA}$ of 1.20. This resulted in a very flat training curve, with the model only slightly improving past the first couple of epochs. Our valence results were fairly good (Figure 1), with lots of tight clustering near the higher values and more spread near the lower values. Our arousal was not quite as good, with the range of predicted values never dipping below 5, while our gold labels went down to 4. It also often predicted a bit higher than the actual value.

Training dataset two was on English restaurant reviews. Our model for this performed very similarly to the English laptop reviews, using a learning rate of $3.5e^{-5}$, dropout of 0.18, and having an $RMSE_{VA}$ of 1.25. The resulting training curve was almost identical, and so were the predictions. However, the predicted valence and arousal on

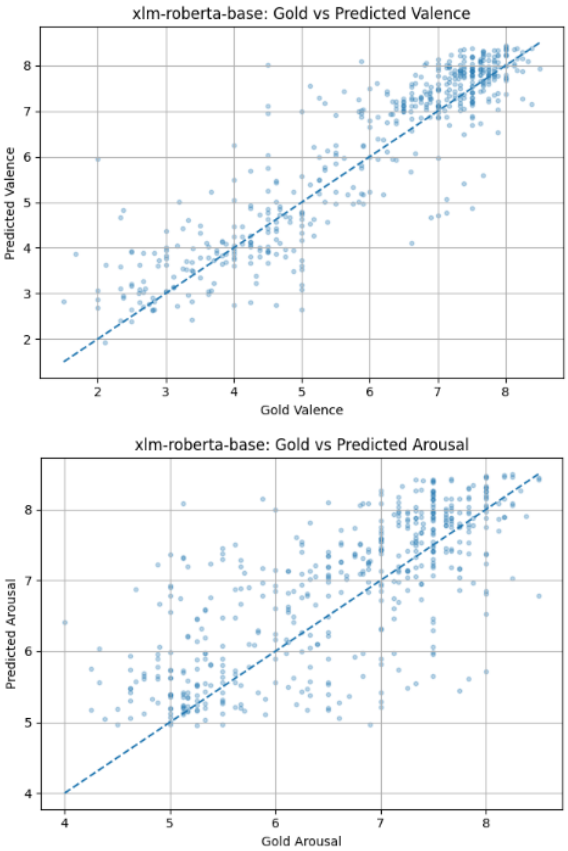

Figure 1: English Laptop Dataset on RoBERTa with Bayesian Optimization

this model were noticeably better. The predictions hugged the line of best fit closer than in the previous model. However, they were also more extreme, with the density overlap being pushed a bit higher than the actual values. It still had the same value range as the previous model, causing it to predict a smaller range of values than the gold labels.

Training dataset three was on Japanese hotel reviews. Our model for this dataset was noticeably worse than the previous one, with some very interesting prediction charts. The learning rate and dropout were quite consistent with our previous model for English laptops, with our best $RMSE_{VA}$ being 0.96. This led to a loss trend that tapered off quite quickly, giving an optimal model by epoch 5. However, this "optimal" model was not very accurate, as our valence scores, while well clustered, had a very small range and centered around 7 or 4, respectively. Our arousal results were even worse, as the model never predicted outside of the range of 6 to 7, while the gold standard labels ranged from 5 to 8.

Training dataset four consisted of Russian restaurant reviews and performed only a little better than

the Japanese hotel dataset. Good learning rates were quite spread out, with $1e^{-5}$ and $5e^{-5}$ having the same impact on the RMSE, which was about 1.41. Using $5e^{-5}$, the loss of the model was somewhat parabolic, decreasing slowly on the way down until hitting its lowest point and starting to climb back up. While not as bad as the Japanese Hotel results, we saw similar clustering within the Valence; however, this model had a larger spread, which led to better results. The Arousal predictions were also only slightly better, with the range of prediction being between 6 and 8, compared to the gold spread of 4 through 8.

Training dataset five was about Ukrainian restaurant reviews, and this was a big step up from previous results. Our optimal learning rate was still $2e^{-5}$, but the dropout was significantly hire than other models at around 0.3, as well as our $RMSE_{VA}$, which was about 1.33. When training, the model consistently improved until 14 epochs and gave us good results. The Valence score was very well clustered with few outliers and the full range of possible results. And the arousal was the best seen so far, with very good clustering around the correct results and a full range of values that matched the range of gold standard values.

Training dataset six was Tatar Restaurant Reviews. This was the worst result of any training in the dataset. The optimization never seemed to converge and left us with a very large spread of possible hyperparameters. It settled on $5e^{-5}$ and a dropout of 0.2 for the model, leading to a very quick training time (3 epochs was the best result) and poor results. This was coupled with a peak $RMSE_{VA}$ of 1.91. Valence was the worst of any training set, not moving outside of the range of 6 to 7.5, while the gold ratings went from 1 to 9. This was even more pronounced in the arousal, where the values ranged from 7 to 7.5, while the actual range was 3 to 9.

Training dataset seven was Chinese Restaurant Reviews. The optimal learning rate sat between $3e^{-5}$ and $4e^{-5}$ with a dropout rate around 0.05. The learning of the model looked much nicer than a lot of other graphs, with loss decreasing up to epoch 9. And this was echoed in our $RMSE_{VA}$, which was about 0.69. This led to much better graphs (Figure 2), with the valence graph having great groupings and accuracy. This led to the model having one of the best valence distributions of any model we tested. The arousal graph was also much better, hitting the full range of potential values and clustering near the correct value. It did skew a little bit higher than the actual value, but was still quite accurate.

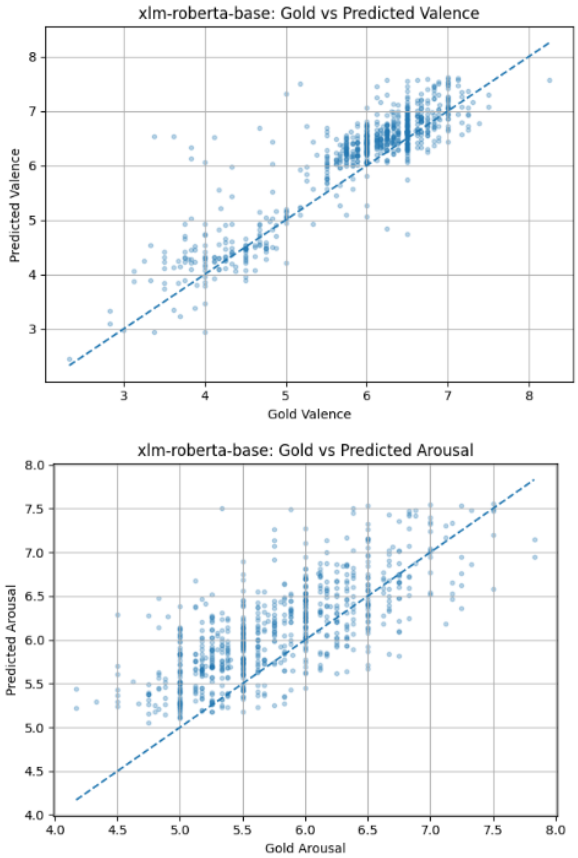

Figure 2: Chinese Restaurant Dataset on RoBERTa with Bayesian Optimization

5.3 Messing Around with Models

While we used Bayesian optimization for the above automatic hyperparameter tuning, we also wanted to see how grid search optimization compared (Figure 3). To test this, we used the English restaurant reviews dataset on the RoBERTA model with grid search to get optimized parameters. This resulted in a learning rate of $2e-5$, dropout of 0.05, and $RMSE_{VA}$ of 1.23. Our results were not as good compared to the Bayesian optimization. The valence was more spread out than in our Bayesian model, leading to only one main cluster and a lot of spread-out results. Arousal also had the range issue seen earlier, where it only spanned 5 to 8 instead of 4 to 9. Overall, this performed worse compared to Bayesian optimization, which aligns with what we expected.

We also ran the English laptop reviews dataset on a new model, mDEBERTa (Figure 4). Just like

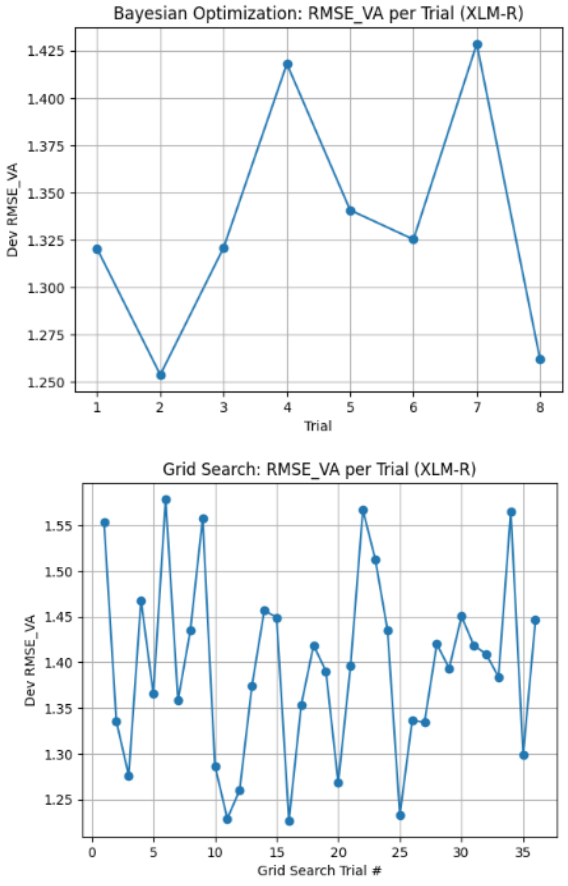

Figure 3: Bayesian Optimization vs Grid Search Optimization in $RMSE_{VA}$ vs trials

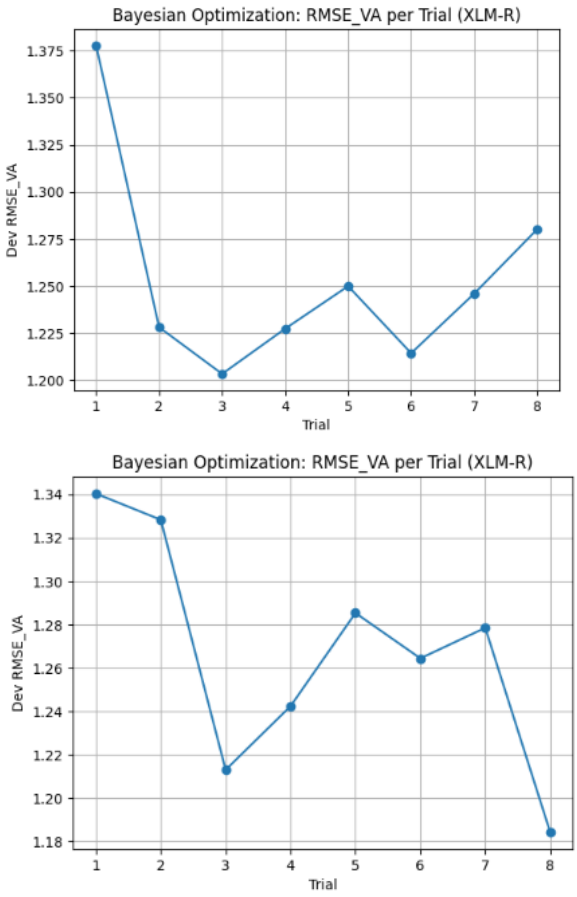

Figure 4: Bayesian Optimization of $RMSE_{VA}$ vs trial on RoBERTa vs mDeBERTa

the above datasets, we ran an automated optimizer to give us a learning rate of $2e^{-5}$, dropout of 0.6, and $RMSE_{VA}$ of 1.18. Compared to our first results, this model performed a bit better, with a better learning curve and predicted values. The Valence was quite good, with a big cluster near the top of the chart which spanned the full range of possible values. The arousal was not as good, with many values being too high and poorly clustered. Compared to our other model, this still performed better.

6 Conclusion

Our experiments overall showed that performance was sensitive to hyperparameter tuning, model architecture, and dataset characteristics. Across the seven datasets, valence consistently achieved a stronger correlation and wider range than arousal, which had an impact from overestimation. The most reliable learning rate was $2e^{-}5$ with dataset-dependent dropouts, which helped prevent overfitting.

Furthermore, mDeBERTa outperformed XLM-RoBERTa on the English Laptop dataset. It improved $RMSE_{VA}$ from 1.20 to 1.18. The gain was small, but consistent, which shows that it captured better subtle sentiment signals that were more relevant to dimensional scoring. This shows that the newer architecture has better improvements for DimABSA. In future works, we would like to experiment more with similar families of BERT.

While evaluating models, it was clear that the Valence often performed the best of the two metrics, with two little groups often forming: one near the top right of the graph and one near the middle, bottom left of the graph. In almost every case (except for the Tatar dataset), the models made these two groups while spreading out data through the rest of the graph. This makes it seem like the models learn to guess more extreme values as opposed to more moderate values for Valence.

For arousal, the story was much different. Only a few models had arousal predictions that spanned the whole range of possible values, with many arousal graphs having distinct ranges for which

they would put a score. The most notable of which was Tatar, with a range of less than 1. When arousal graphs looked good, they spread the data out quite well, but often over-predicted the value by a little bit. In many of the graphs, it can be seen that most of the predicted values are above the line of best fit. However, it was often quite close to the actual result, which made the arousal graphs seem accurate when there was a good data spread.

References

Sven Buechel and Udo Hahn. 2017. EmoBank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, pages 578–585, Valencia, Spain.

Saif M. Mohammad and Felipe Bravo-Marquez. 2017. WASSA-2017 shared task on emotion intensity. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA)*, pages 34–49, Vancouver, Canada.

Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. 2020. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pages 8600–8607, New York, USA.

Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval)*, pages 486–495.