BASTE: BERT, CNN, and T5 for **A**nalysis of **S**entiment in a **T**ext **E**nsemble Learning Model

Cody Burker, Dustin Cox, Margo Suryanaga

Abstract

Digitization trends spanning nearly every industry have given rise to ever-growing textual data in the form of customer reviews, emails, chat support, and more. Understandably, organizations have a desire to effectively measure sentiment associated with this data, and design effective responses in turn. The rise of Natural Language Processing (NLP) techniques in the domain of sentiment analysis have been robustly documented in academic literature, pushing the boundaries with regard to state-of-the-art performance. In our approach, we join the recent forays into taking an ensemble approach in an effort to further extend the state-of-the-art combining individual approaches (BERT, CNN, and T5) into various ensemble approaches (rule-based, SVM, KNN, logistic regression, a neural network, and decision tree). Results indicate that effective ensemble approaches may be harder to achieve than anticipated given only one (rule-based) was able to marginally outperform the best individual model (BERT).

Keywords: Sentiment analysis; natural language processing; ensemble modeling;

Introduction

Understanding the sentiment among one's customer base, particularly with regard to the products and services offered by an entity - whether a company, non-profit, or government - is a key capability in a broader strategy to scaling effectively, detecting defects or deficiencies in offerings, and responding quickly to issues. Furthermore, as digitization trends continue to accelerate in nearly every industry, such capabilities are no longer "nice-to-have," but increasingly are required to keep pace with competitors and meet the evolving expectations of customers.

Specific use cases for an enhanced ability to detect sentiment from customer-submitted text vary greatly across domains. A software-as-a-service company may desire to deploy such a sentiment classifier over all incoming support chat submissions, quickly prioritizing limited resources to address the most unhappy customers; a not-for-profit offering mental health services may need to understand the most severe cases in their queue-based on text messages received through their text-support offering in order to rapidly deploy specific professionals (e.g., a psychiatrist vs. counselor) to patients; and a government division may benefit from being able to quickly categorize public comments into logical in-favor or opposed-to groupings during a public comment period, thereby accelerating permitting processes and reducing the cost of legally-required stages of a public works project.

Our goal in this project was to improve upon best-in-class sentiment analysis techniques by combining constituent models in an ensemble approach which could predict a 5-star rating based on textual reviews submitted by customers.

Background

BERT is one of the pre-trained language models known to have improved results when carrying out sentiment analysis. Hoang et al., discovered that a model using BERT to carry out aspect-based sentiment analysis performed better than one without BERT[1]. Improved BERT predictions are seen as a direct result of its bidirectionality, as it allows the model to consider a word's context from the left and right directions concurrently. As such, we chose BERT as one of our base constituent models.

As Kim discovered in experiments on the use of CNN for both classification and sentiment analysis, CNN models are able to achieve superior results with a simple implementation and little hyperparameter fine-tuning[2]. In part due to this, CNN models are one of the most commonly used architectures for sentiment analysis[3]. Keeping this in mind, we chose to build a CNN model as our second base model.

The use of T5 for sentiment analysis is a fairly new approach but is being explored due to some of the unique features of the T5: 1) The multi-tasking aspect of T5 allows for sentiment analysis and classification to be performed concurrently 2) The T5 model allows for a greater amount of tokens as compared to BERT at times driving better accuracy 3) Its nature as both an encoder and decoder automatically generates the output as a text format as opposed to BERT[4]. Thus, we chose to use a T5 model as our third base model.

Ensemble learning is an approach that is known to yield improved results compared to its components. It is a way that researchers have found will allow for the mitigation of the limitations of base models. Shahri et al. have used an ensemble approach to identify domain-specific comentions of proteins and phenotypes in papers, combining BERT, a CNN, and an RNN into a stacked model which outperformed the constituent model[5]. Yang et al. also formulated an ensemble model to perform sentiment analysis on e-commerce product reviews using a combination of a sentiment dictionary, BERT model, CNN model, BiGRU model, and attention mechanism[6].

---

[1] Hoang et al., "Aspect-Based Sentiment Analysis Using BERT" https://aclanthology.org/W19-6120.pdf
[2] Kim, Yoon. "Convolutional Neural Networks for Sentence Classification" https://aclanthology.org/D14-1181.pdf
[3] Wang et al., "Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model" https://aclanthology.org/P16-2037.pdf
[4] Chebolu et al., "Exploring Conditional Text Generation for Aspect-Based Sentiment Analysis" https://arxiv.org/pdf/2110.02334.pdf
[5] Shari et al., DeepPPPred: An Ensemble of BERT, CNN, and RNN for Classifying Co-mentions of Proteins and Phenotypes
[6] Yang et al.,"Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning" https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8970492

Methods

Data

We identified the Yelp Dataset[7], which consisted of roughly 7 million customer textual reviews labeled with 5-star scale ratings (1 being most negative and 5 being most positive), as an ideal data source for this investigation. The size and quality of this dataset, coupled with the fact that it had been widely used for academic and educational purposes in the Natural Language Processing domain, undergirded our confidence. Understanding we would be training three base models (CNN, BERT, and T5) at Level 0, and then a plethora of ensemble approaches at Level 1, we anticipated needing a total of 180,000 random examples which would be divided into three partitions:

1. *Train Set 1 (see Figure 1):* 80,000 would be used to fit the base models at Level 0.
2. *Train Set 2 (see Figure 2)*: 80,000 would be run through each of the Level 0 base models to predict the 5-star rating, and these predictions would then be passed to the ensemble models to fit at Level 1.
3. *Test Set (see Figure 3)*: 20,000 data points would be used to measure the accuracy of each base and ensemble model in order to benchmark performance.

Models

Base Models (Level 0)

The following models were fitted on Train Set 1 and evaluated predictions for Train Set 2 as well as base accuracies on Test Set used for comparison purposes against the Ensemble Models (see Figure 4 for model architecture):

1. *CNN:* We used a simple tokenizer, trained on Train Set 1 (for initial training and evaluation, but also predictions on Train Set 2 and Test data) padded and truncated to 1,024 tokens. We used a model with an embedding layer, a filter and max pooling layer, and a dropout layer with dense layers inspired by Brownlee[8]. This performed fairly well considering the simplicity and comparatively small number of parameters. Parameters were tuned (e.g. dropout, filters), and we also experimented with starting with pre-trained word2vec embeddings instead of learning our own embedding, which slightly decreased accuracy, although the model did train in significantly less epochs (<20 epochs instead of ~60 epochs). We also experimented with training on a more balanced training set but this slightly reduced accuracy.
2. *BERT:* We used the ‘bert-base-cased’ tokenizer, available pretrained, in order to tokenize the textual examples in Train Set 1 with a max length of 512 tokens (the max BERT supports). We padded shorter examples to the max, and truncated those which exceeded the max length, giving us Input IDs, Token Type IDs, and Attention Masks for the 80,000 examples which were fed into the ‘bert–base-cased’ pretrained model. The BERT model’s output was then passed to a densely-connected hidden layer utilizing a ReLU activation function, which in turn fed into a densely-connected classification layer

[7] https://www.yelp.com/dataset
[8] Brownlee, Deep Convolutional Neural Network for Sentiment Analysis (Text Classification)

utilizing a sigmoid activation function to normalize the predicted probabilities for each of the possible 5 classes. For this classification layer we utilized the Adam optimizer and Sparse Categorical Cross Entropy loss, which had the benefit of helping us avoid 1-hot encoding the star ratings. This model performed quite well after just 3 epochs, each of which took between 1.5-2 hours. Given the high relative performance, computational resource constraints, and Google Colab Pro+ timeout risks, we did not attempt additional epochs beyond the 3 to keep training times manageable to within a day as we iterated through tuning.

3. *T5:* We applied the Simple Transformers T5 model pre-trained on the t5-base corpus. As part of fine-tuning the model, we chose to set a maximum text length of 512 limiting the tokens to those we saw contained the most prominent signals. We also chose to set the number of training epochs to 2 after testing model performance with up to and including 3 epochs. We did not attempt additional epochs beyond the 3.

Ensemble Models (Level 1)
The following models were fitted on predictions for Train Set 2 (outputs of the Base Models above) and used to generate final predictions on Test Set:

4. *Logistic Regression:* We trained 300 logistic regression models utilizing a range of learning rates, maximum iterations, and solvers - both with and without class weighting. The best performing model did not use class weights, used the standard 'lbfgs' solver, a learning rate of .01, and capped convergence iterations at 100. To train the models, we first utilized simple class predictions from each of the constituent models (i.e., 1-5), but including the confidences associated with those class predictions from BERT and CNN improved performance overall.
5. *SVM:* We trained an SVM on the predictions, which performed decently with a radial basis function (RBF) kernel but performed worse with a linear kernel. However, the radial basis function does not scale well to datasets even of this size (80,000 rows) which posed a challenge in tuning the model.Training on the BERT and CNN confidences (instead of just the categorical prediction) yielded a higher accuracy, however normalizing the T5 prediction (to be between 0 and 1 instead of 1-5) actually lowered the accuracy.
6. *Decision Tree:* We trained 800 decision trees utilizing Gini and Cross Entropy loss, random vs. best splitters, and various levels for splitting criteria and minimum examples for each leaf in the tree. The best performing model utilized Cross Entropy loss, a random splitter, a minimum of 10 examples to split, and at least 7 examples per leaf.
7. *Neural Network:* We created a simple 1-layer multi-layer perceptron (mlp) with 100 neurons and an L2 regularization term of 100, which resulted in poor performance.
8. *K-Nearest Neighbors:* We tuned a KNN model and found that using 25 neighbors resulted in a decently performing model using the confidences of the base models.
9. *Rule-Based:* We implemented a rule-based model using a custom set of rules fine-tuned using Train Set 2. The rules first take into account commonly generated predictions. Should all three models generate the same prediction then that will become the final predicted value of the ensemble model. Similarly, if two out of three models evaluated

the same prediction then ideally we would use that predicted value. However, in this case we additionally took into account the confidences of the predictions from each model to see if the final predicted value should instead revert to the least commonly generated prediction. Should all three models have differing predictions, we once again take into account the confidences of the predictions of each model to see if that is the predicted value which should be used. We believe that this is a fairly good indicator if the predicted value is accurate based on our initial analysis of the correlation between confidence and accuracy on the Train Set 2 data (see Figure 5).

Results and discussion

	Model	**Accuracy**
Baseline	Most Common Class (5)	43.98%
Reference Models	Naive Bayes	50.61%
	Random Forest	45.58%
Base Models	BERT	72.51%
	CNN	66.46%
	T5	65.16%
Ensemble Models	Rule-Based	72.52%
	Support Vector Machine (SVM)	71.95%
	K-Nearest Neighbors	71.77%
	Logistic Regression	71.61%
	Neural Network	71.45%
	Decision Tree	69.98%

Table 1. Accuracy results of base and ensemble models on Test Set

Accuracy results of running the base and ensemble models on the Test Set are presented in Table 1 above. In addition, Table 1 contains the accuracy evaluated from a baseline (picking the most common class) as well as two reference models (Naive-Bayes and Random Forest) trained with a term frequency representation of the data (limited to the most common 5,000 terms due to memory limitations).

Both the Naive-Bayes and Random Forest reference models yielded relatively low accuracies, but were still higher than the baseline. These reference models performed significantly worse than the more sophisticated base and ensemble models, in alignment with expectations given

the state-of-the-art status among the base models we selected. The base models also had the benefit of being fine-tuned whereas the reference and baseline models were simple models.

Per the accuracies in Table 1, we see that the ensemble models consistently outperformed both the CNN and T5 base models, but failed to materially improve accuracy beyond that observed from the BERT base model. When reviewing the predictions on the Test Set from the rule-based ensemble model, we found that all three of our models agreed in predictions majority of the time: 50% of the accurate predictions and 39% of the inaccurate predictions (see Figures 6 and 7). Additionally, over 95% of the predictions on the test set had at least two out of the three models agree on the same prediction.

Conclusion

Although we were not able to beat the BERT base model, our ensemble models at least outperformed the CNN and T5 base models. One note is that when we binned the data into positive, neutral and negative (4-5 stars, 3 stars, and 1-2 stars, respectively) the rule-based model accuracy jumped 17% - while our model was correct 72.51% of the time on exact prediction of the star rating, when considering these bins it was correct 89.36% of the time. Moreover, of the incorrect predictions all three of our models agreed on the incorrect prediction 39% of the time, suggesting that this was the optimal prediction given the training data, as seen in Figure 6.

Another possible explanation for the lack of performance in our ensemble is poor calibration of the models. Since we attempted to exploit the probability provided by the CNN and BERT output layers by selecting the output of the model that is more confident in its prediction, if the models' confidence were not related to how well it performed, it could cause poor performance in our ensemble models. However when exploring the accuracy of the models compared to their confidence (measured as the highest probability of all the predictions) in Figure 5, the models appear to be at least fairly well-calibrated given accuracy increases with the confidence.

Given more time, we would suggest training the models as if the training data were continuous (instead of categorical) to explore whether this allows models to capture the ordinal relationships between categories better. It would also be interesting to inspect the accuracy of a model that performed within plus or minus 1 star of the actual record, in order to see if the models are wildly incorrect or just marginally so. We would also suggest looking at the STSB task within the T5 model which was pre-trained to classify text to a floating point number on a 1 to 5 scale and takes into account 21 different sentiment classes.

References

1. Shari et al., DeepPPPred: An Ensemble of BERT, CNN, and RNN for Classifying Co-mentions of Proteins and Phenotypes
2. Hoang et al., “Aspect-Based Sentiment Analysis Using BERT” https://aclanthology.org/W19-6120.pdf
3. Brownlee, Deep Convolutional Neural Network for Sentiment Analysis (Text Classification)
4. Yang et al.,”Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning” https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8970492
5. Kim, Yoon. “Convolutional Neural Networks for Sentence Classification” https://aclanthology.org/D14-1181.pdf
6. Wang et al., “Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model” https://aclanthology.org/P16-2037.pdf
7. Chebolu et al., “Exploring Conditional Text Generation for Aspect-Based Sentiment Analysis” https://arxiv.org/pdf/2110.02334.pdf

Appendix

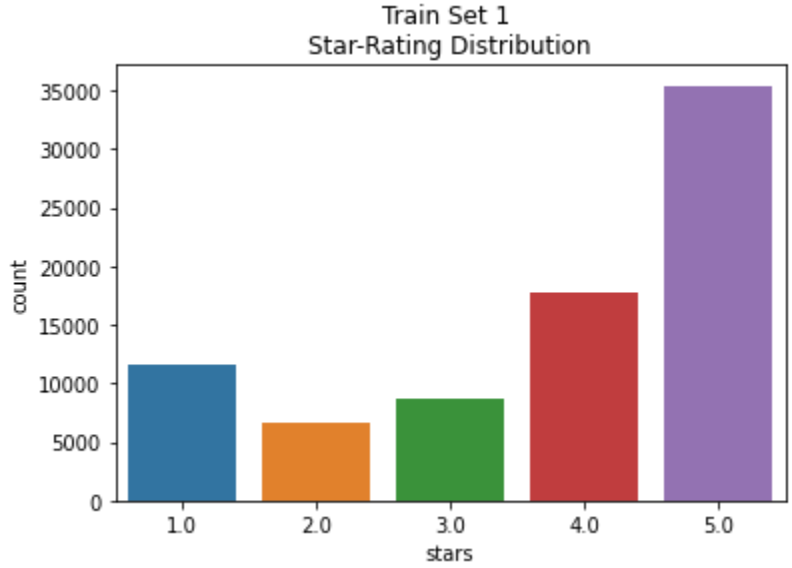

Figure 1. Star-rating distribution on Train Set 1

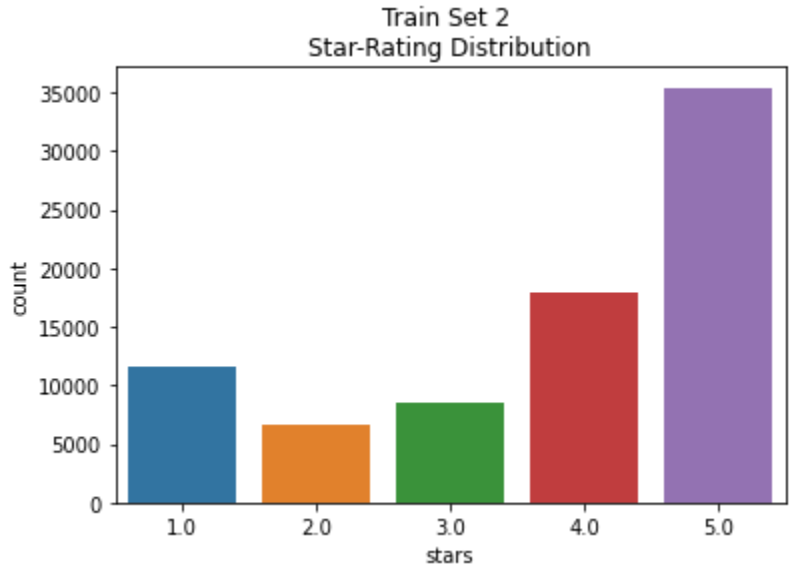

Figure 2. Star-rating distribution on Train Set 2

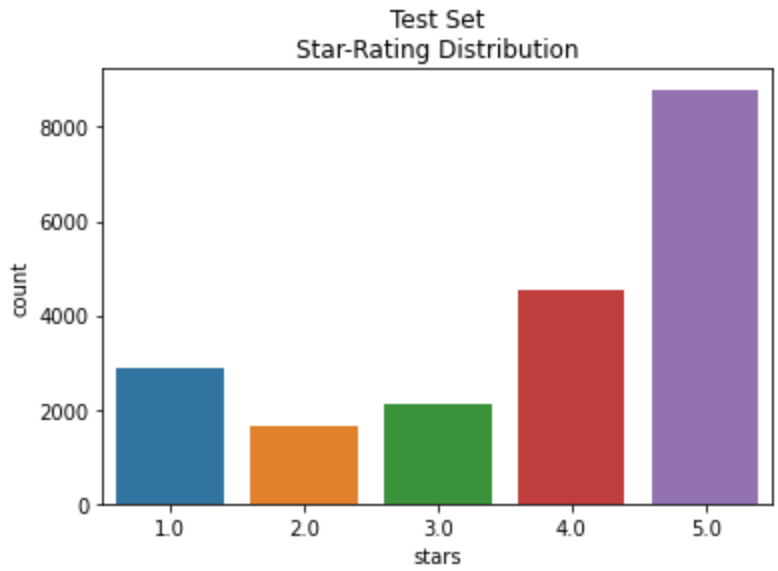

Figure 3. Star-Rating Distribution on Test Set

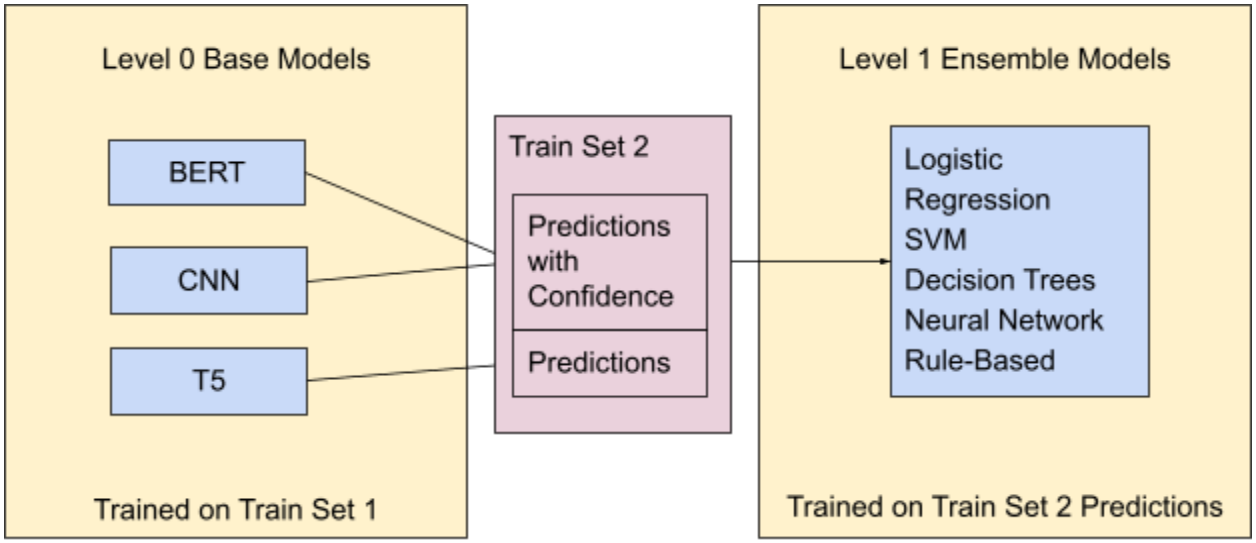

Figure 4. Architecture of Ensemble Models

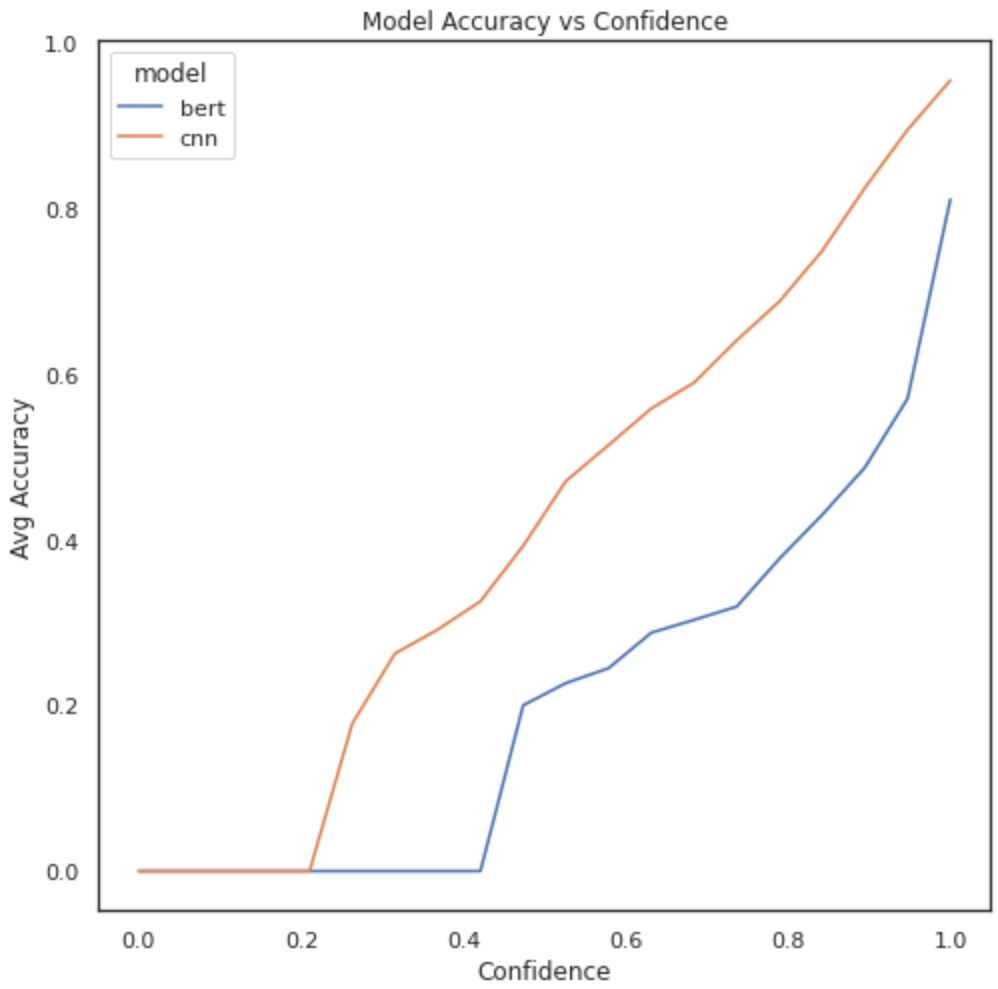

Figure 5. Comparison of Confidence (Highest class probability) vs Accuracy on Train 2

```
All three match:  12200
Two matches - BERT & CNN:  2592
Two matches - BERT & CNN Revert to T5:  2
Two matches - BERT & T5:  2254
Two matches - BERT & T5 Revert to CNN:  0
Two matches - CNN & T5:  88
Two matches - CNN & T5 Revert to BERT:  1935
One match - BERT:  928
One match - CNN:  0
One match - T5:  1
Remaining:  0
Val accuracy:  0.72515
```

Figure 6. Summary of prediction/rule used within Rule-based Ensemble Model on Test Set

Prediction used	
All three match	2135
BERT	530
BERT & CNN	942
BERT & CNN Reverted to T5	2
BERT & T5	858
CNN & T5	59
CNN & T5 Reverted to BERT	970
T5	1

Figure 7. Summary of prediction/rule used within Rule-based Ensemble Model on Test Set which yielded inaccurate results