Predicting Customer Satisfaction Using Call Transcripts and Deep Learning Models

Jason Papale
papale47@ischool.berkeley.edu

Abstract

Understanding customer satisfaction is of critical importance to the long-term success of a business. For companies which interface with their customers primarily through call centers, there is an opportunity to better understand customer satisfaction by leveraging telephone transcripts and deep learning modeling techniques. This paper presents the findings and methods associated with three deep learning modeling approaches which were evaluated for this purpose using data from a financial services company. Each of the approaches represents a different level of complexity, which was seen as important in determining the optimal model architecture given the domain of the data and quality of transcripts available. The first and least complex approach entailed utilizing a Convolutional Neural Network (CNN) to predict customer satisfaction given an entire transcript. The second approach entailed breaking each transcript into shorter segments and then fine-tuning a Bidirectional Encoder Representations from Transformers (BERT) model to predict customer satisfaction given a single transcript segment. The third approach entailed Hierarchical Transformers, where either a Long Short-Term Memory (LSTM) model or a Transformer was utilized to predict customer satisfaction given a sequence of segment-level embeddings obtained from the fine-tuned BERT model. Of these methods, the CNN approach performed the best, achieving an average precision-recall score of .30 for the "unsatisfied" class and an overall F1 score of .64.

1 Introduction

One of the primary methods through which companies that operate call centers gauge the satisfaction of their customers is through an after-call survey. While this method does provide relevant customer satisfaction information to companies, there are several limitations associated with it. First, only a small fraction of callers actually take this survey. This can result in excessively wide confidence intervals when conducting statistical analyses on this kind of data, especially when assessing a specific subset of customers, as is often the case with call center analytics. A second limitation pertains to the ability to generalize the results of after-call surveys to the entire customer base. For example, it is entirely possible, if not probable, that there are certain kinds of customers which are more likely to participate in an after-call survey than others. If these differences in customers are sufficiently substantial, then the after-call survey volume which is available may not be representative of the entire customer base, which could, in turn, lead to an inaccurate understanding of customer satisfaction as a whole. One solution which has the potential to address both of these shortcomings is the application of deep learning techniques which are able to utilize the entirety of every transcribed conversation between a customer and the telephone representative.

Though customer satisfaction can be evaluated along a number of dimensions, the dimension which received the most focus as part of this effort centered around a customer's overall satisfaction with the company's ability to satisfactorily complete his or her request on that call. Compared to other dimensions of customer satisfaction, such as a customer's satisfaction with the last representative he or she spoke to, this dimension of customer satisfaction is likely to be more latent and, consequently, present more of a challenge to deep learn-

ing models. For example, it is likely that in many instances, the conversation between the customer and the representative is very cordial, yet the customer could still end the call feeling dissatisfied for reasons that were not explicitly discussed during the call, such as a the customer having to wait in queue for an extended period before speaking the representative. That said, this dimension of customer satisfaction was chosen given that, from a business perspective, it is one of the most important.

In terms of the criteria that were used to evaluate the effectiveness of the various models, one of the main metrics which was used was the average precision-recall score for the "unsatisfied" class. This is because, much like fraud detection models, being able to identify problematic situations, which, in this case, is an unsatisfied customer, is of higher importance from a business standpoint because it enables the business to pursue corrective action. While the specific recall and precision parameters which would be required for the model to be deemed successful would ultimately be decided by the business and its evaluation of the cost of implementing corrective measures based on model predictions, for the purposes of this effort, success was gauged based on improvement in the average precision-recall score for the "unsatisfied" class relative to the baseline precision-recall score which could be achieved without a model (.15). Specifically, any model which achieved an average precision-recall score of .30 for the "unsatisfied" class was considered successful, as it would represent a 100% increase over baseline. Additionally, an average-precision score of this level has the potential to be considered sufficient for at least some low-cost business initiatives aimed at correcting a poor customer experience (e.g. sending a followup text to the customer to confirm the dissatisfaction and attempt to rectify the situation/make amends).

2 Related Work

Before discussing the details of the models employed as part of this effort, there are several other related bodies of work which were referenced that should be mentioned. First, Zhong & Li (2019) demonstrated success utilizing CNNs with call center transcripts from an auto dealership to classify caller intent into one of four pre-determined categories. Of note, the authors detail using the entirety of the transcript but removing the portion of the conversation specific to the telephone representative, so as to allow the model to focus only on customer's speech. This not only reduced the amount of data being fed to the model and associated time needed to train it, but it also improved model performance. This modification seemed at least as equally reasonable in a customer satisfaction context, where the signal is much more likely to be in the customer's text, and the part of the transcript specific to the telephone representative is comparatively more likely to be noise. For this reason, the telephone representative portion of the transcript was removed as part of the modeling efforts detailed in this paper. It is also worth noting that while the CNN model implemented by Zhong & Li (2019) achieved superior performance to that which was achieved by the models described herein, one likely reason for this has to do with the fact that a customer will generally verbalize the reason he or she is calling (i.e. intent), whereas the degree to which a customer is satisfied is not as likely to be explicitly discussed. In fact, in many conversations, it is reasonable to expect that a customer's satisfaction could change over the course of the call, making it even more difficult for a model provide an accurate classification.

Another pertinent body of work which was referenced as part of this effort was that of Pappagari et al. (2019). In their paper *Hierarchical Transformers for Long Document Classification*, the authors describe what they refer to as Hierarchical Transformers. This type of architecture entails stacking either an LSTM (Hochreiter & Schmidhuber, 1997) or Transformer (Vaswani et al., 2017) on top of another Transformer model. More specifically, the authors describe first decomposing lengthier transcripts into smaller segments and then using BERT (Devlin et al., 2019) to produce segment-level embeddings which can be stacked into a sequence. These embeddings are then ingested by either an LTSM or a Transformer which ultimately performs the classification. If an LSTM is used on top of BERT, the authors refer to this model architecture as Recurrence over BERT (RoBERT). If a Transformer is used, the authors refer to the architecture as Transformer over BERT (ToBERT). One of the main limitations that the authors are able to overcome with their Hierarchical Transformer methodology is the limitation around sequence length that Recurrent Neural Networks (RNNs) and Transformer models suffer

from. Thus, by breaking documents into smaller segments, the authors were able to process longer text while still leveraging BERT and its ability to capture long-range dependencies within sequence of text and providing a single embedding which effectively captures the meaning of that segment. When the authors took the additional step of fine-tuning BERT as discussed in *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* by Delvin et al. (2019), they were able to achieve accuracy levels of greater than 80% on document classification tasks. Of particular interest, one of the datasets on which the authors trained and evaluated RoBERT and ToBERT models centered around customer satisfaction in a call center context. While the dimension of customer satisfaction centered around the telephone representative as opposed to the overall experience, it still provides a good benchmark against which to compare the performance of the models implemented as part of this effort.

3 Methodology

Three modeling approaches were implemented in order to determine the optimal model architecture given the domain of the data and quality of transcripts available. All three models utilized the same dataset, which consisted of 82,898 transcripts, each labelled with an after-call survey score. To facilitate training, evaluating, and formally testing the model, 62,141 of these transcripts were randomly selected as training records with 6,925 and 13,832 being randomly assigned to the development and testing groups, respectively. Additionally, as mentioned above, the telephone representative portion of each transcript was removed at the onset in order to allow the models to better focus on the part of the conversation where the signal for customer satisfaction is most likely to reside. It is also worth mentioning that the dataset was fairly imbalanced where most transcripts where labelled as "satisfied." Additionally, since greater than 95% of the transcripts (after removing the telephone representative portion) were less than 1500 words, all transcripts were truncated to this length.

In order to determine the optimal hyperparameters to use in the different models, a subset of 7000 records from the training data was created so that the model could be evaluated across a range of hyperparameters. Of note, each iteration of model hyperparameters was evaluated for 2 epochs in order to also observe the rate at which the model was able to learn given the selected hyperparameters. After the hyperparameter evaluation routine was complete, a table was generated for each model which included the training and validation losses, as well as the validation accuracy for each different combination of hyperparameters. The specific selection of hyperparameters was based on observed improvement in loss and accuracy over epochs, initial loss and accuracy levels, and trade-offs in complexity versus gains in model performance. The tabulated results of the hyperparameter evaluation routine, as well as the reasons for hyperparameter selection, are included in the code associated with this effort.

Once the optimal hyperparameters were identified, the models were then trained on the full training dataset. The CNN model was trained for 15 epochs, the BERT fine-tuned models trained for 3 epoch, and the ToBERT and RoBERT models were both trained for 10 epochs. These epoch amounts were sufficient to observe indications overfitting for all models. After training was complete, the model weights associated with the training iteration that had the lowest loss with respect to the validation dataset were saved and used in final model evaluation against the testing dataset. This process ensured that the models were trained in an optimal way with respect to hyperparameters, as well as to overfitting. Lastly, cross-entropy was used as the loss function. Of note, as discussed in Madabushi et al. (2019), the option to insert weights into the loss function in order to more heavily penalize an incorrect prediction with respect to the minorty class and help mitigate the effects of the imbalance in the dataset was utilized.

3.1 Convolutional Neural Network

The first modeling approach which was used in this effort was that of a CNN. In this approach, the transcripts were tokenized using the Keras tokenizer and then padded to the maximum length of 1,500 words. The primary elements of model architecture consisted of an embedding layer, three convolutional and max-pooling layers, and two fully connected layers. The ReLu activation function was used for the first fully connected layer, which contained 1,000 dimensions, and the Softmax activation function was utilized for the second fully connected layer, which contained 100 di-

mensions and functioned as the output layer. Additionally, a dropout layer (p=.4) was utilized after the concatenation of the max-pooled convolutional layers, as well as after the first fully connected layer in order to minimize overfitting. Each of the convolutional layers consisted of 200 filters each, and kernel sizes of 3, 4, and 5 were used for the first, second, and third convolutional layers, respectively. Lastly, 100-dimensional GloVe embeddings were utilized in the embedding layer.

3.2 BERT Fine-Tuned

The second model that was implemented was a BERT fine-tuned model. As discussed above, the first step in the process to fine-tune a BERT model on the transcripts was to break the transcripts into smaller segments. In this case, the segments consisted of 200 tokens each and a 50 token overlap between adjacent segments in order to provide some degree of contextual continuity between segments. The fine-tuning was accomplished by taking the CLS token generated by the BERT model for each segment and then feeding it into two fully connected layers. The first fully connected layer had 768 dimensions, such that the output of this fully connected layer (which also included a BatchNorm layer and the ReLu activation function) would be of the same number of dimensions as the CLS token. The second fully connected layer then functioned as the output layer and was followed by a Softmax activation function. Of note, a search for optimal hyperparameters for the fine-tuned BERT model was not performed for two reasons. The first had to due to the BERT model taking an excessive amount of time to train and evaluate. The second reason stems from BERT fine-tuning guidance where, according to Delvin et al. (2019), "large data sets (e.g., 100k+ labeled training examples) [are] far less sensitive to hyperparameter choice than small data" (p. 14). For that same reason, the model was only trained for 3 epochs. Of note, while BERT's built in token embedding layer was used, the positional embedding layer was not, as it was not found to have a significant effect on model performance in a similar context (Pappagari et al., 2019).

3.3 Hierarchical Transformers

The third model approach entailed the use of Hierarchical Transformers, which, in this case, amounted to an extension of the fine-tuned BERT model. Specifically, once the BERT model was fine-tuned, it was used to produce segment-level embeddings which were then stacked into a sequence that could be fed into either a Transformer (ToBERT) or LSTM (RoBERT). Of note, since some documents contained more segments than other, documents with less than the maximum number of 10 segments were padded, where each padded segment consisted of a 768-dimensional vector of zeros. Additionally, the Adam optimizer was used in both ToBERT and RoBERT.

3.3.1 ToBERT

The Transformer portion of the ToBERT model included 2 encoder layers, where each layer consisted of a 6-headed self-attention sub-layer and a 2,048-dimensional fully connected sub-layer. The decoder portion of the Transformer was not used, since the benefit of the Transformer can be obtained solely with the encoder layer for classification tasks. Since the encoder outputs a vector for each token embedding, which in this case is an embedding for a transcript segment, an average pooling layer was added following the output of the encoder to condense the ten output vectors into one vector. This 768-dimension output vector could then be considered a document-level embedding. Following the transformer layer, the 768-dimension vector was inputted into a 30-dimensional fully connected layer with a subsequent BatchNorm layer and ReLu activation function before arriving at the output layer with Softmax activation.

3.3.2 RoBERT

The LSTM portion of the RoBERT model consisted of 2 recurrent layers (i.e. a stacked LSTM), with each layer having 100 hidden dimensions. Utilizing a bi-directional LSTM, as well as adding more hidden dimensions was evaluated as part of the hyperparameter evaluation process; however, neither appeared to improve model performance in a significant way. The output layer and a dropout layer (p=.4) followed after the LSTM layer, followed then by the Softmax activation function.

4 Results

The results of model performance with respect to the F1 score for the "unsatisfied" class, the overall F1 score, and overall accuracy can be seen in Table 1.

Model	F1-Unsatisfied	F1-Overall	Accuracy
CNN	30%	64%	58%
BERT - FT	27%	74%	71%
ToBERT	25%	55%	48%
RoBERT	25%	50%	43%

Table 1: Summary of Model Performance

Based on the results, it is clear that the CNN outperformed (in terms of the average precision-recall score for the minority class) both the fine-tuned BERT model, as well as the Hierarchical Transformer models. Also, although the CNN only achieved an average precision-recall score of .30 for the minority class and a 64% overall accuracy, this was achieved using the model weights obtained after only the second epoch. These weights were selected because the validation loss began to increase after the second epoch, as can be seen in Figure 1. However, after 15 epochs, the average precision-recall score for the minority class only fell to .25, but the overall accuracy increased to 86%, representing a substantial improvement within that metric. The average precision-recall for the majority class also rose from 70% to 92%. Further, based on a baseline precision recall of .15, the CNN's performance meets the aforementioned success criteria and may be sufficient to have some application.

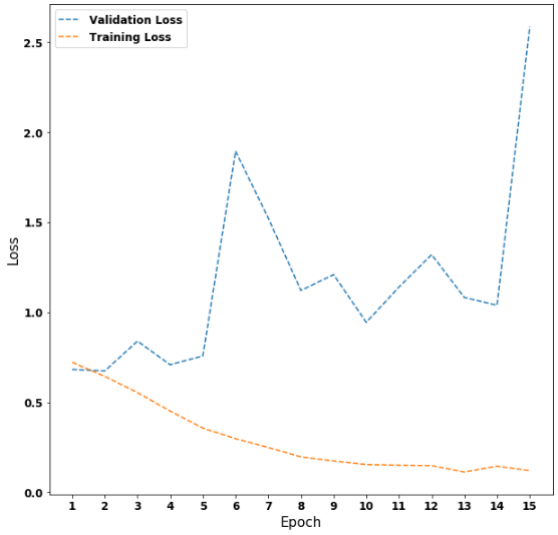

Figure 1: Loss Over Epochs - CNN

While the fine-tuned BERT model did slightly under-perform the CNN model with respect to the average precision-recall score the minority class, it outperformed with respect to overall F1 (74%) and accuracy (71%). As with the CNN (and as can be seen in Figure 2), after the second epoch of model training, validation loss began to increase, though not to the same extent as with the CNN model, while training loss steadily decreased.

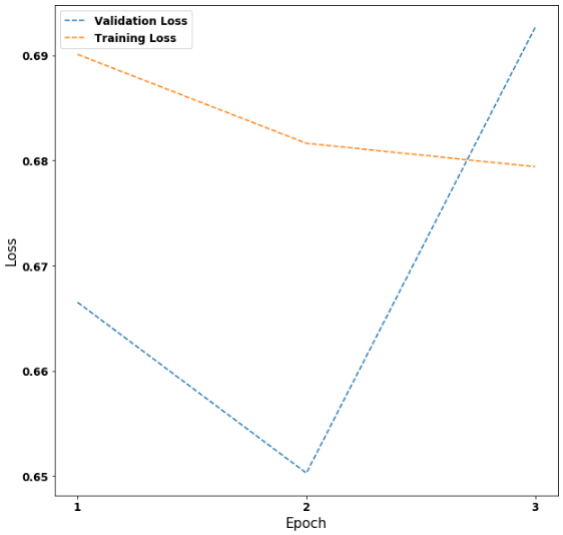

Figure 2: Loss Over Epochs - BERT Fine-Tuned

The ToBERT and the RoBERT models performed similarly, with both under-performing the CNN and fine-tuned BERT models. Specifically, the ToBERT and RoBERT models both achieved an average precision-recall score of 25% for the "unsatisfied" class and an overall accuracy of 48% and 43%, respectively. The loss trends observed for ToBERT and RoBERT during training can be seen in Figures 3 and 4, respectively.

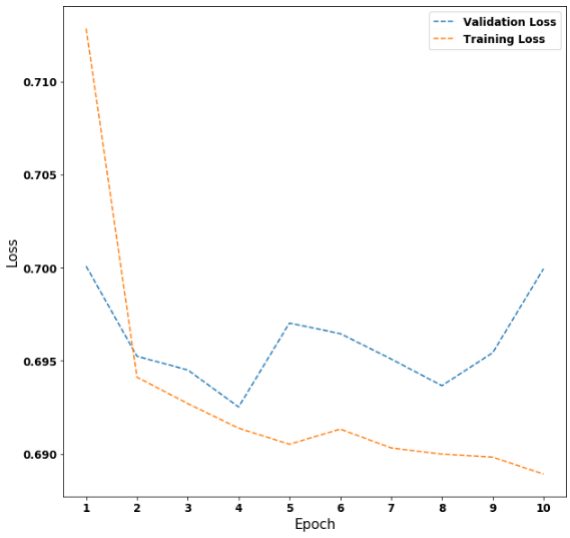

Figure 3: Loss Over Epochs - ToBERT

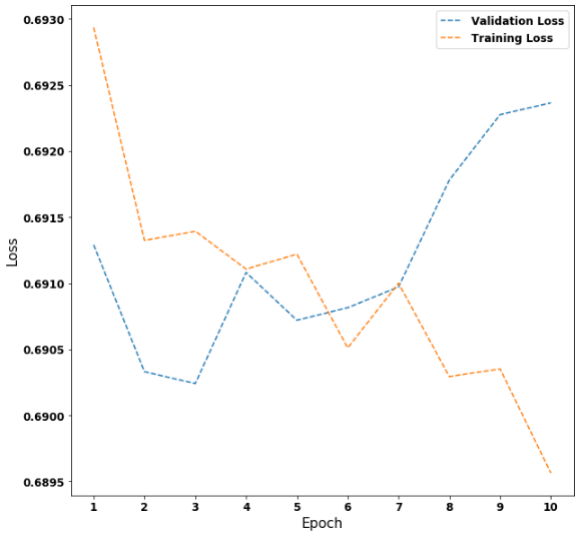

Figure 4: Loss Over Epochs - RoBERT

When the precision-recall curves of all of the models are viewed together (Figure 5), it is clear that the CNN outperforms all of the other models; however, it is by a narrower margin with fine-tuned BERT model. Of note, the ToBERT and RoBERT models do not appear to show much improvement in precision until a recall of 0 is almost reached, which is not the case with the CNN and BERT fine-tuned models.

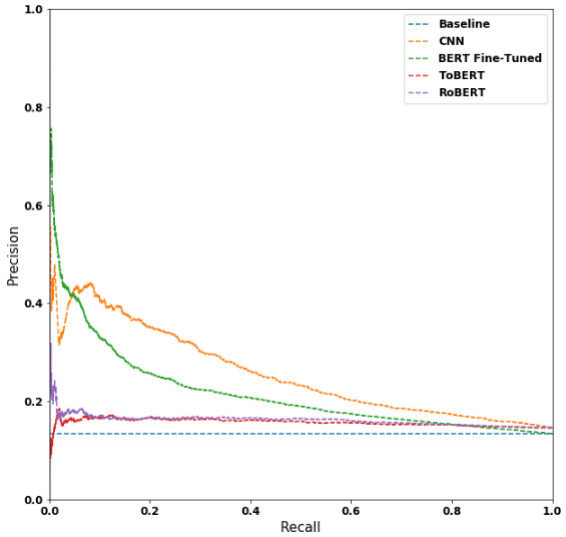

Figure 5: Precision-Recall Curves

5 Conclusion

Based on the findings of other researchers, the expectation was that ToBERT and RoBERT would outperform a fine-tuned BERT, which would outperform a CNN. The results of this study conclude the exact opposite, with the CNN outperforming all other models among the primary metric of interest. One possible reason as to why this occurred may have had to do with the fact that the telephone representative portion of each transcript was removed. While this method was shown to be effective for a CNN model (Zhong & Li, 2019), removing chunks of the conversation may have been detrimental to the other modeling techniques which were designed to process entire sequences of text, resolve long-range dependencies, and build a temporal context within the sequence. Another possible reason is that the speech within the transcripts was often times fairly fragmented. In some instances, this was due to portions of the conversation being redacted for security reasons. In other instances, it was simply due to inaccuracies in the transcription. For these reasons, it seems reasonable that the CNN model, which has a smaller aperture when assessing the text (i.e. 3-5 words), could outperform the more sophisticated models. It would also seem to follow that if the transcripts were sufficiently fragmented that the benefits typically gained by fine-tuning a BERT model on the text could not be realized, that any models building on top of BERT (i.e. RoBERT and ToBERT) might perform even more poorly.

References

[Zhong & Li 2019] Zhong, J., Li, W. 2019. *Predicting Customer Call Intent By Analyzing Phone Call Transcripts Based on CNN for Multi-Class Classification.*

[Pappagari et al.2019] Pappagari, R., Zelasko, P., Villalba, J., Carmiel, Y., Dehak, N. 2019. *Hierarchical Transformers for Long Document Classification.* Automatic Speech Recognition and Understanding Workshop, 2019.

[Devlin et al.2019] Devlin, J., Ming-Wei, C., Lee, K., Toutanova, K. 2019. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.*

[Hochreiter & Schmidhuber 1997] Hochreiter, S., Schmidhuber, J. 2017. *Long short-term memory.*

[Vaswani et al.2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., Polosukhin, I. 2017. *Attention Is All You Need.*

[Madabushi et al.2019] Madabushi, H., Kochkina, E., Castelle, M. 2019. *Cost-Sensitive BERT for Generalisable Sentence Classification with Imbalanced Data.*