Predicting Customer Satisfaction with Soft Labels for Ordinal Classification

Etienne Manderscheid, Matthias Lee
Dialpad Canada Inc.
1100 Melville St #400
Vancouver, BC, Canada, V6E 4A6
{etienne,matthias.lee}@dialpad.com

Abstract

In a typical call center, only up to 8% of callers leave a Customer Satisfaction (CSAT) survey response at the end of the call, and these tend to be customers with strongly positive or negative experiences. To manage this data sparsity and response bias, we outline a predictive CSAT deep learning algorithm that infers CSAT on the 1-5 scale on inbound calls to the call center with minimal latency. The key metric to maximize is the precision for CSAT = 1 (lowest CSAT). We maximize this metric in two ways. First, reframing the problem as a binary class, rather than five-class problem during model fine-tuning, and then mapping binary outcomes back to five classes using temperature-scaled model probabilities. Second, using soft labels to represent the classes. The result is a production model that supports key customer workflows with high accuracy over millions of calls a month.

1 Introduction

1.1 Motivation

Call centers have been using CSAT surveys to measure Customer Satisfaction for decades. Like most CSAT surveys, those our company provides are delivered either on the line at the end of a call or in response to an SMS message. In the survey, the customers are asked to rate their customer service experience on a 1-5 scale, with 1 being very dissatisfied and 5 being very satisfied, respectively. However, we found that for a typical call center, only up to 8% of callers leave a CSAT survey response[1]. Since only a small fraction of customers leave a survey response, managers and coaches of traditional call centers are missing important information. Specifically:

1. The mean CSAT score suffers from response bias, as customers with a strongly positive or negative experience are far more likely to take the time to respond (Table 1).
2. When the customer has a sub-optimal experience but does not leave any feedback, the call center may not be able to proactively take necessary actions in a timely manner to improve customer experience.

To address these issues, we have developed and present here an algorithm that infers CSAT scores on call center calls with high accuracy and low latency. At the moment of writing, our predictive CSAT feature is fully deployed at scale and has rated over 50 million calls.

1.2 Intended Uses of predictive CSAT

At our company, predictive CSAT is used for coaching purposes, to maintain and improve overall customer experience and to create new opportunities for analytics.

In “Coaching Hub” we provide coaches with material for both recognition of agents and improvement in the form of two lists, with calls rated with predicted CSAT scores of 5 and 1 respectively. Therefore, the precision of classes 5 and 1 is critical - it’s necessary that the calls in these lists are reliably satisfied and dissatisfied, respectively. Since satisfied calls outnumber dissatisfied calls by a wide margin, the precision of class 1 has long been our limiting factor and therefore our primary focus.

Maintaining and improving overall customer experience is crucial for our users. Users such as call center managers, coaches, and agents use predicted CSAT to proactively identify dissatisfied customers moments after the call ends by reviewing calls with predicted CSAT scores of 1 or optionally 2. This enables users to follow up with customers and potentially save their accounts.

Predicting CSAT also creates new opportunities for analytics. We examine which factors are most

[1] for the 691 call centers in our dataset with at least 50 CSAT survey responses, the 10th and 90th percentile of survey response rates were 0.3% and 8%, respectively.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 5: Industry Track, pages 652–659
July 10-12, 2023 ©2023 Association for Computational Linguistics

associated with dissatisfied calls, where the predicted CSAT score is 1 or 2. For example, we might report that calls associated with hold times longer than 10 minutes are associated with a higher percentage of dissatisfied calls in a call center. In this way, we offer data-driven recommendations to improve CSAT for this call center. In interviews with our target users (call center coaches and managers), we learned that a certain amount of error tolerance is acceptable when inferring CSAT scores. Consequently for these 3 applications, it's acceptable if the model predicts a 1 and the customer left a survey response of 2. This motivates our introduction in the System Overview section of precision* and recall* which are metrics with an error tolerance of 1.

1.3 Constraints

Our solution space is constrained in the following ways:

- High precision of class 1: As discussed above in section 1.2, this metric is our primary focus, being central to Coaching Hub's lists of coachable calls and being the minority class relative to class 5. Thus, it is important for our model to have high precision in predicting calls for the lowest CSAT class.

- Latency: The predicted CSAT score is included in a call summary that is shown to the user 10 seconds after the call, and availability within 10 seconds at least 99.9% of the time is a hard requirement for all features displayed in the summary. This holds even if the transcript is many thousands of words long. Since we deploy this model on CPU to control cost and availability, this was non-trivial to achieve.

The typical, out-of-the-box deep learning solution for solving multiclass classification problems is to allow each distinct label as a possible output of the neural network, and train using a loss function such as cross-entropy loss over the set of all labels. As a shorthand, we refer to this approach here as "5-way classification". However, this approach does not lend itself well to meeting constraints 1 and especially 2.

The main contribution of this paper is to present a combination of two techniques, which were adapted for this problem to solve both constraints:

CSAT Rating	Number of labels
1	39k
2	9k
3	8k
4	17k
5	222k
Total	**296k**

Table 1: The CSAT survey distribution favors the extremes. This phenomenon is well known in the contact center space and is explained by reporting bias: since taking a survey takes time and effort, customers that are strongly motivated by a very positive or very negative experience are more likely to leave a response than customers with a relatively normal experience

1. **Binary classification + fan-out:** We reframe the 5-class prediction problem as a binary classification task during model training and then map temperature-scaled model probabilities back from 2 to 5 classes during inference time ('fan out').

2. **Soft labels:** We introduce a modified label smoothing approach that achieves superior accuracy for this ordinal classification task.

2 Related Work

There are few research studies on predicting customer satisfaction (CSAT) scores on contact center conversations using transcripts generated by an Automatic Speech Recognition (ASR) model. In Bockhorst et al., 2017, they developed a system that not only utilizes call transcripts transcribed by an ASR model but also other non-textual data such as call duration, queue, in-queue waiting times, utterance level sentiment scores, and various customer data. Overall, there are 5,501 features in the training dataset. The author's model is trained to predict a metric called Representative Satisfaction Index (RSI) which is the average of four different survey scores. In the end, their framework involves two models, namely a rank scoring and an isotonic regression model. In a more recent study, Auguste et al., 2019 used the Net Promoter Score (NPS) to predict customer satisfaction on chat conversations. A promoter score can be defined as a rating that customers give to indicate how likely they are to promote a company. Out of a scale of 0 to 10, customers with ratings of 9 or 10 are considered promoters whilst those with ratings of 0 to 6 are considered detractors. NPS is calculated as the difference between promoters and detractors and companies want this metric to be positive and as high as possible. They compared macro F1 scores across different classification methods and their best method yielded a macro F1 score of 53.8%, which they noted that is a rather limited performance. Other studies that looked at predicting CSAT on contact center conversations proposed using information extracted from raw audio signals such as acoustic, emotions, and prosodic features. (Park and Gates, 2009; Zweig et al., 2006; Vaudable and Devillers, 2012; Devillers et al., 2010)

Contact center managers are usually interested in picking out calls with a low CSAT score for either coaching purposes or for identifying opportunities to take meaningful interventions in a timely manner to improve customer experience. Hence, it's important to identify these calls with a relatively high degree of precision. Label smoothing is a regularization technique introduced by Szegedy et al., 2016 that has been successfully used to improve accuracy of the Inception architecture on the ImageNet dataset. In Müller et al., 2019, it is noted that label smoothing has been adopted in training procedures of other state-of-the-art image classification models (Zoph et al., 2017; Real et al., 2018; Huang et al., 2018). In another domain such as speech recognition, Chorowski and Jaitly, 2016 used label smoothing to reduce word error rate on the WSJ dataset. Additionally in machine translation, Vaswani et al., 2017 was able to slightly improve the BLEU score

3 System Overview

3.1 Dataset

We only used the call transcripts as input to the model. The transcripts are produced by our company's proprietary Automatic Speech Recognition (ASR) models. This simplifies model deployment and helps latency as the model can be run as soon as a transcript is available, without waiting for any additional features, and it was sufficient to obtain high accuracy. We then preprocessed transcripts of contact center conversations to create training, validation and test sets.

The labels collected were CSAT survey responses left by customers. Labels were aggregated into a single dataset rather than many separate company-specific datasets. Surveys were either run at the end of the call ("please stay on the line for a brief survey...") or sent to customers as an SMS message. Survey responses have a customer satisfaction (CSAT) rating of 1-5, where 5 is the highest satisfaction. Table 1 shows the distribution of CSAT customer ratings over our dataset.

Additionally, we excluded callers that were present in the training set from the validation and test sets[2] to prevent contamination of these sets [3].

3.2 Model Fine-tuning

We used the Big bird[4] model hosted on Huggingface[5](Wolf et al., 2020) for all experiments. We chose Big bird (Zaheer et al., 2020) as our model architecture because it is a transformer-based model capable of handling long sequences (up to 4096 tokens) with low latency in our production environment. Specifically, its memory requirements scale linearly in the number of tokens rather than quadratically as many transformers-based models do. If transcripts exceeded 1536 tokens[6] in length, only the last 1536 tokens of the conversation were used and the preceding were discarded; this occurred in 16% of transcripts. This allowed us to keep latency and cost under control at inference time.

We trained all models using cross-entropy loss and a learning rate of $10e{-5}$ with early stopping. The metric we chose to evaluate the checkpoints is motivated by the user experience around CSAT, as detailed in section 1.2.

As a result, the precision of class 1 is more important than the precision of other classes, or than recall, and an error tolerance of 1 is acceptable for our intended use cases. Therefore, we introduce the metric "precision*", i.e., the precision with an error tolerance of 1. We also formally define a modified version of true positive and false positive (denoted tp* and fp* respectively)[7] which is necessary in

[2] The final size of the validation and test sets were 2996 and 2943 calls, respectively

[3] We also excluded the data of one company from the validation and test set because its CSAT distribution was so unusual (99% of responses were 1s and 2s) we suspect a misconfiguration of the survey for that company.

[4] https://huggingface.co/google/bigbird-roberta-base

[5] https://huggingface.co/

[6] tokens: a part of a sentence, usually a word, but can also be a subword (non-common words are often split in subwords) or a punctuation symbol

[7] Where CSAT$_p$ denotes the predicted CSAT, CSAT$_s$ denotes the CSAT survey response, and class $c \in (1, 2, 3, 4, 5)$. As an example using class c=2, TP2* is the count of predicted CSAT = 2 where survey $CSAT \in (1, 2, 3)$

defining precision*.

$$\mathrm{TP}_{\mathrm{c}}^{*} = |\mathrm{CSAT}_s \in (c \pm 1) \land \mathrm{CSAT}_p = c| \quad (1)$$

$$\mathrm{FP}_{\mathrm{c}}^{*} = |\mathrm{CSAT}_s \notin (c \pm 1) \land \mathrm{CSAT}_p = c| \quad (2)$$

$$\mathrm{FN}_{\mathrm{c}} = |\mathrm{CSAT}_s = c \land \mathrm{CSAT}_p \neq c| \quad (3)$$

$$\mathrm{Precision}^{*} = \frac{\mathrm{TP}^{*}}{\mathrm{TP}^{*} + \mathrm{FP}^{*}} \quad (4)$$

$$\mathrm{Recall}^{*} = \frac{\mathrm{TP}*}{\mathrm{TP}^{*} + \mathrm{FN}} \quad (5)$$

For the purposes of picking the best model checkpoint for any experiment, we measure the F-beta metric with $\beta = 0.5$ on class 1 with a tolerance of 1 and define it as follows:

$$F_{\beta}^{*} = \frac{(1+\beta^2) \times \mathrm{Precision}^{*} \times \mathrm{Recall}^{*}}{(\beta^2 \times \mathrm{Precision}^{*}) + \mathrm{Recall}^{*}} \quad (6)$$

Using beta = 0.5 achieves our goal of weighing both precision and recall while giving precision more importance than recall.

3.3 Binary Classification + Fan-Out

First, we mapped the CSAT labels to the 0-1 range. For example, when using hard labels, the CSAT label vector $[1, 2, 3, 4, 5]$ is remapped to $[0, 0, 0, 1, 1]$. Since the remapped vector contains 2 classes, we can train a binary classifier. At inference time, we first rescale the fine-tuned model output, logits with temperature scaling. Temperature scaling simply divides the logits by a single parameter that is fitted on a held-out validation set so the model probabilities are better calibrated (Guo et al., 2017). Typically, for a classification task, the logits from a model are passed through a softmax function to get final class probabilities. Instead, we use the low-CSAT class probability to "fan-out", i.e. we map this probability $\in [0, 1]$ back into 5 classes using 4 class thresholds. For example, if the class 5 threshold is 0.15, then a model probability (of low CSAT) of 0.01 corresponds to a 5, while a model probability of (of low CSAT) 0.16 maps to a 4. We illustrate this fine-tuning and inference method in Figure 1.

The algorithm we devised to infer the class thresholds is the following. As explained previously, class 1 $F_{0.5}^{*}$ is our primary metric. Thus we set the class 1 threshold first in a way to optimize it. Specifically, we use a loop to search the parameter space of class 1 threshold values and pick the one that optimizes class 1 $F_{0.5}^{*}$ on the validation set. Then we repeat this process with the thresholds that separate classes 5-4, 2-3, and 4-3 (the priority is determined by user workflows) to optimize class 5 $F_{0.5}^{*}$, class 2 $F_{0.5}^{*}$ and then class 4 $F_{0.5}^{*}$. Once the 4 thresholds are set, the 5 classes are defined by them. Here's a real example of threshold values: $[0.92, 0.45, 0.09, 0.03]$, and these separate classes 1-2, 2-3, 3-4 and 4-5 respectively.

3.4 Soft Labels

The traditional label smoothing equation is

$$y_k^{LS} = y_k(1-\alpha) + \alpha/K \quad (7)$$

where K is the number of label classes, y_k is a one-hot encoded label vector and α is the hyperparameter that determines the amount of smoothing (Müller et al., 2019).

Our motivation for trying label smoothing is the ordinal nature of CSAT classes. That is, a call with a survey response of '2' is a low CSAT call, but not as strongly as a '1', and more strongly so than a '3'. So when using binary classification as detailed above, it's natural to try label smoothing with a higher level of smoothing for the center classes. We also refer to labels that have been smoothed as "soft" labels. In Table 2 we show the values of α we used for different classes, reserving the weakest α for the outermost classes (1,5) and the strongest α for the center class (3). Aside from using these multiple values of α , we implemented label smoothing to train the model in the standard way.

CSAT Class	α	Soft Labels
1	0.02	0.99
2	0.2	0.90
3	1	0.5
4	0.2	0.1
5	0.02	0.01

Table 2: Different smoothing values were applied to each of the 5 CSAT classes that resulted in the soft labels used for training

3.5 Experiments

We conducted a total of 24 experiments, each exploring a different permutation of the experimental conditions. The conditions are shown in Table 3.

This setup allowed us to explore multiple experimental conditions while generating variability for statistical analysis but limiting the number, cost and carbon footprint of our experiments.

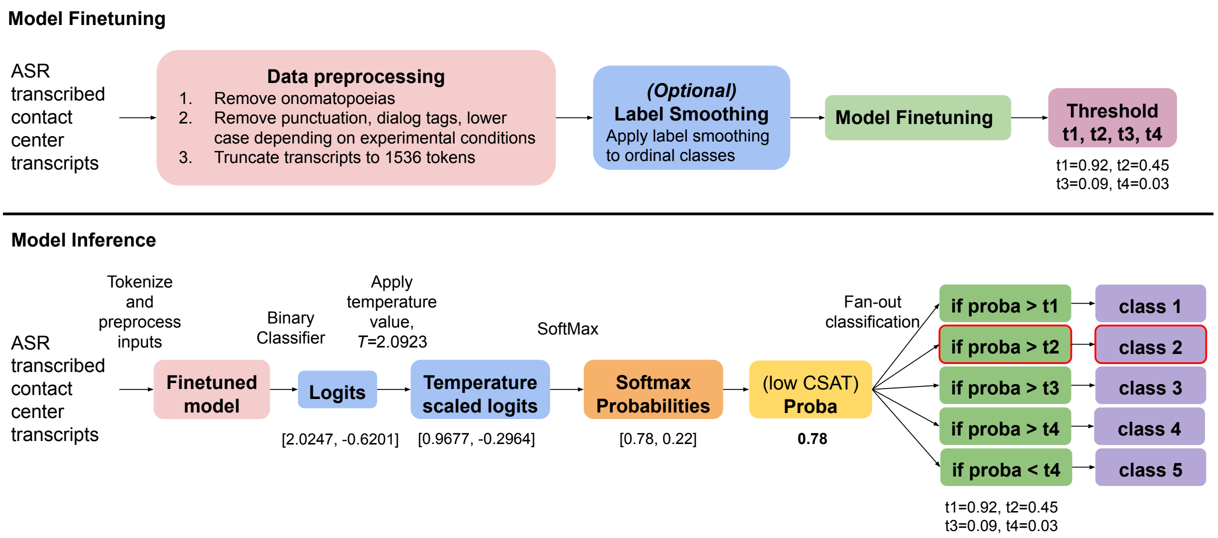
Model Finetuning
ASR transcribed contact center transcripts
Data preprocessing
1. Remove onomatopoeias
2. Remove punctuation, dialog tags, lower case depending on experimental conditions
3. Truncate transcripts to 1536 tokens
(Optional) Label Smoothing
Apply label smoothing to ordinal classes
Model Finetuning
Threshold t1, t2, t3, t4
t1=0.92, t2=0.45 t3=0.09, t4=0.03
Model Inference
ASR transcribed contact center transcripts
Tokenize and preprocess inputs
Finetuned model
Binary Classifier
Logits
[2.0247, -0.6201]
Apply temperature value, T=2.0923
Temperature scaled logits
[0.9677, -0.2964]
SoftMax
Softmax Probabilities
[0.78, 0.22]
(low CSAT) Proba
0.78
Fan-out classification
if proba > t1
if proba > t2
if proba > t3
if proba > t4
if proba < t4
class 1
class 2
class 3
class 4
class 5
t1=0.92, t2=0.45 t3=0.09, t4=0.03

Figure 1: The overall training and inference method used

Experimental Condition	Cardinality	Set
Type of Classification	3	Binary Soft Labels / Binary Hard Labels / 5-way classification
Punctuation	2	Included / None
Dialog Tags	2	Included / None
Casing	2	Lowercase / Uppercase

Table 3: Our 24 experiments corresponded to every permutation of these experimental conditions

4 Results and Discussion

4.1 Type of Classification

On precision and "precision*" of classes 1 and 5, soft labels binary classification performed best (figure 2 and table 4. All 8 t-tests of binary soft labels vs the other two have p-values < 0.05). Furthermore, within the binary classification + fan out approach, soft labels worked better than hard labels on almost every metric. It had higher precision on every class (most important for our users), and better on recall for $\frac{3}{5}$ classes.

5-way classification had the highest precision on center classes by a wide margin. It also produces the strongest recall for classes 1 and 5, probably because the binary classification approaches optimize for precision of these 2 classes the most. In terms of overall accuracy (with a tolerance of 1), binary soft labels and 5-way classification were statistically tied (90.4% vs 90.6% respectively), with binary hard labels trailing slightly (89.2%).

4.2 Conclusion

In this paper, we propose an approach to maximize the precision of certain classes in the context of an ordinal classification problem. We show that for our application it makes sense to cast the problem first as binary classification and restore the 5 output classes using probability thresholds. We also show that the use of soft labels outperforms that of hard labels in our setup. This approach can benefit applications where ratings can be formulated as ordinal classes and where some classes are emphasized over others in the primary user workflows. We also show that the problem of CSAT prediction is amenable to modern deep learning techniques with high accuracy using the transcript as the sole input to the model.

5 Limitations

- We use only the transcript as input to the model. This implies the model wouldn't know that a hold was long unless the customer said "that was a long hold" or something to that effect. The transcript usually contains language indicating the hold is taking place "may i place you on hold?", "thanks for holding", etc, but rarely indicates the exact duration of the hold. Similarly, the model doesn't know the wait time unless the customer complains explicitly about it.

Metrics	Binary Hard labels	Binary Soft labels	5-way-classification
Class 1 Precision*	77.9%	**83.4%**	78.0%
Class 2 Precision *	57.3	63.8	**82.9%**
Class 3 Precision *	23.7	35.9	**79.7%**
Class 4 Precision *	78.5	80.3%	**94.9%**
Class 5 Precision *	94.5%	**95.3%**	92.5%
Class 1 Precision	68.5%	**74.2%**	69.5%
Class 2 Precision	5.8%	5.6%	**19.4%**
Class 3 Precision	6.7%	11.7%	**50.5%**
Class 4 Precision	12.4%	14.1%	**55.0%**
Class 5 Precision	88.5%	**89.7%**	86.7%
Class 1 Recall	58.0%	55.5%	**65.2%**
Class 2 Recall	7.7%	**10.7%**	3.1%
Class 3 Recall	6.4%	13.2%	**14.4%**
Class 4 Recall	9.2%	**15.4%**	11.9%
Class 5 Recall	91.7%	90.7%	**96.9%**

Table 4: Precision*, precision and recall for each class

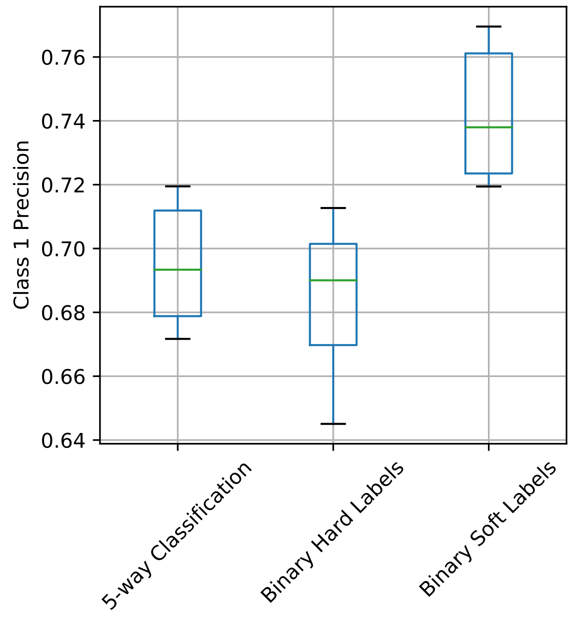

Figure 2: Class 1 Precision as a function of Classification Type

- Predicted CSAT is available 10 seconds after the end of the call. New applications become possible if predicted CSAT is made available continuously throughout the call since a manager would be able to "whisper" (advise the agent on the call without the customer hearing), message or barge (jump into the call as a 3rd party). An important concern if predicted CSAT is computed repeatedly will be managing the cost and carbon footprint, possibly by using a small model as an initial gating function.

- If a call center doesn't collect CSAT surveys through our company, their accuracy will be impacted as they won't be reflected in the training or test set. We ensure customers understand this by training our agents to explain it and including it in help center documentation.

6 Ethics Statement

- We have read and abide by the ACL Code of Ethics [8].

- **Data Privacy**: We follow the data privacy measures in place at our company which include scrubbing personal identifiable information (PII) from customer data and restricting our use of customer data to improvements to the services we provide them. We did not rely on any external annotations.

- **Intended Use by Customers**: In the product we highlight both high and low CSAT calls for review by a supervisor to ensure employees receive a mix of positive and constructive feedback. Since supervisors review calls, they can adjust incorrect classifications produced by the model.

[8] https://www.aclweb.org/portal/content/acl-code-ethics

- **Potential bias:** We sample subpopulations of users and their customers and evaluate internally to ensure the model outputs are not biased against specific groups.

- **Carbon Footprint:** We minimized the carbon footprint of our experiments while meeting the need for variability required by statistical analysis. We achieved this by running 24 experiments, each with different experimental conditions, rather than running multiple experiments with different random seeds within each of the 24 conditions. In total the experiments described in this paper represented less than 500 hours of computation on a single V100 GPU.

7 Acknowledgements

We thank personA for setting up some data tables used to generate some of the analyses presented here, and personB for his idea to use CSAT classes 2, 3 and 4 for lower confidence predictions. We also thank personC and personD for advice on preprocessing and model training, and personE for help reviewing this paper.

References

Jeremy Auguste, Delphine Charlet, Geraldine Damnati, Frederic Bechet, and Benoit Favre. 2019. Can we predict self-reported customer satisfaction from interactions? pages 7385–7389.

Joseph Bockhorst, Shi Yu, Luisa Polania, and Glenn Fung. 2017. Predicting self-reported customer satisfaction of interactions with a corporate call center. In *Machine Learning and Knowledge Discovery in Databases*, pages 179–190, Cham. Springer International Publishing.

Jan Chorowski and Navdeep Jaitly. 2016. Towards better decoding and language model integration in sequence to sequence models. *CoRR*, abs/1612.02695.

Laurence Devillers, Christophe Vaudable, and Clément Chastagnol. 2010. Real-life emotion-related states detection in call centers: a cross-corpora study. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 2350–2353. ISCA.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. *CoRR*, abs/1706.04599.

Yanping Huang, Yonglong Cheng, Dehao Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V. Le, and Zhifeng Chen. 2018. Gpipe: Efficient training of giant neural networks using pipeline parallelism. *CoRR*, abs/1811.06965.

Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. 2019. When does label smoothing help? *CoRR*, abs/1906.02629.

Youngja Park and Stephen C. Gates. 2009. Towards real-time measurement of customer satisfaction using automatically generated call transcripts. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, page 1387–1396, New York, NY, USA. Association for Computing Machinery.

Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V. Le. 2018. Regularized evolution for image classifier architecture search. *CoRR*, abs/1802.01548.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Christophe Vaudable and Laurence Devillers. 2012. Negative emotions detection as an indicator of dialogs quality in call centers. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5109–5112.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. *CoRR*, abs/2007.14062.

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. 2017. Learning transferable architectures for scalable image recognition. *CoRR*, abs/1707.07012.

G. Zweig, O. Siohan, G. Saon, B. Ramabhadran, D. Povey, L. Mangu, and B. Kingsbury. 2006. Automated quality monitoring for call centers using speech and NLP technologies. In *Proceedings of* *the Human Language Technology Conference of the NAACL, Companion Volume: Demonstrations*, pages 292–295, New York City, USA. Association for Computational Linguistics.