

Automating Managers' Responses to Hotel Reviews

Stanford CS224N Custom Project
Mentor: Alexandre Matton

Surya Ierokomos
Graduate School of Business
Stanford University
siero@stanford.edu

Xiangbing Ji
Center for Professional Development
Stanford University
xji1994@stanford.edu

Sandy Lee
Department of Management Science and Engineering
Stanford University
sl269@stanford.edu

Abstract

Firms have strong incentives to manage and maintain their reviews on online platforms as they have been shown to have a causal impact on their revenues [1]. For large firms, it can become prohibitively expensive in terms of time and money for managers to manually decide which reviews are worth responding to and then to compose the appropriate response. The goal is to create a deep learning system which takes firm characteristics and specific review text as an input and outputs potential responses to be posted on the review platform. In this paper, we aim to resolve this bottleneck by building a BERT Sequence Classifier to trigger a sentiment specific Seq2Seq model to generate automated responses only in appropriate contexts. Our model produces sensible review responses with Rouge-1 F1 scores varying from 0.28 to 0.33 on average during tests. This has the potential to be meaningful to the firm as it could reduce labor costs and make customers feel like their voice is being heard on these platforms. Likewise, this could be meaningful to consumers because it would allow a greater chance of their review being responded to.

1 Introduction

The rapid advancement of technology has led to an increase in the number of consumer review platforms which allow consumers to express their opinions to products by writing reviews and giving ratings. These websites have continuously gained traction over the last few years and serve as an integral source of information for consumers [1]. Such platforms have been especially crucial for hotel organizations, as understanding consumers' needs and wants is imperative for their business as they compete with the bulk of rivals offering similar experience and functionalities. Hotel reviews on platforms like TripAdvisor act as a direct communication channel between hotels and customers, and analysis on these reviews can help hotel managers to discover and address issues and defects in a timely manner. Empirical studies indeed highlight that responding to customer feedback in an accurate and timely manner improves customer experience as well as business performance [2]. In spite of these benefits, with the ever-increasing number of reviews received daily, it has become all the more difficult and expensive for managers to manually respond to reviews. This illustrates the importance of automatic response generation, which will be the focus of our work.

Neural text generation is one of the most active fields in deep learning, and it is an indispensable natural language processing task which involves building a neural network based model trained with a set of text token sequences. This learned model can then be used to generate texts relevant to the training data. However, building an intelligent system to automatically generate suggested replies to

reviews is not a standard machine learning problem. We hypothesize that state-of-the-art pre-trained high-performance language models can be used to generate more semantically meaningful text. In our approach, we apply such models to generate high-quality management responses to hotel reviews.

Furthermore, there is a high cost to misgenerated sentiment predictions in this setting. For example, outputting the most typical response observed in our data, the phrase “Thank you”, to a negative review is highly undesirable. Such a response would likely antagonize the customer. Therefore, there is the added complexity of getting the sentiment and tone of the response to align with the respective review. In other words, when a review is positive, the response should sound positive and appreciative. When the review is negative, the response should sound apologetic to some degree. We call this problem the *misgenerated sentiment problem*.

We hypothesize that an out-of-the-box application of OpenNmt would suffer from this misgenerated sentiment problem. We believe this would be the case because: 1. The modal response in our data is “Thank you”, leading to a high possibility of a predicted response of “Thank you” to a negative review. 2. The baseline application would not use the extra contextual information of star ratings. To combat this problem, we propose a trigger module to activate our context specific language generation task.

2 Related Work

In the field of natural language processing, dialogue generation has been studied extensively to facilitate social conversations [3]. In addition, researchers have more recently seen an increase in interest in implementing smart reply and chatbot systems especially for commercial platforms [4] [5] [6]. Such work generally leverages the Neural Machine Translation Model (basic RNN Encoder-Decoder Model) [7] which is an end-to-end learning approach for automated translation. For instance, Google’s Smart Reply system [4], which generates automated response suggestions for e-mails, is grounded in the Neural Network Model with long short-term memory (LSTM) [8] trained on large-scale email conversations. In the model, the context and corresponding responses are treated as source and target sentences respectively. Rather than generating free text, Google’s Smart Reply uses a semi-supervised approach to generate a response pool from which it selects the final response based on the LSTM predictions. Likewise, LinkedIn uses a statistical model for recommending replies to messages[9].

Elsewhere, NMT has been applied to a variety of software engineering tasks such as automatically producing commit messages based on code changes [10] and generating a sequence of APIs when provided with a natural language query [6]. While all such tasks are highly relevant, the application of the NMT model for response generation to hotel reviews on online platforms has not been evaluated. We attempt to fill such gap. There are, however, some fundamental differences between review-response dialogues and previous NMT tasks like Smart Reply. Importantly, hotel reviews and managers’ responses are generally one round of dialogue, and at the same time, the limited context from short reviews make it more difficult to generate a concise, relevant response. This suggests that it might be appropriate to take into account lengths of input sentences before directly applying the NMT model to hotel reviews.

3 Approach

3.1 Preprocessing

As the dataset was scraped directly from the website and as we delved deeper into the data, we realized it required significant preprocessing. Namely, we stripped non-english and special characters from our dataset. As a final step to our cleaning, we manually checked the data quality over random subsamples before we proceeded to the main model training. We had to remove a couple vitriolic and antagonistic reviews to maintain the integrity of our language generation task. This was particularly time intensive and was a bottleneck in our pipeline.

3.2 Baseline

To our knowledge, we are the first to study generated language in this domain. Therefore, there is no gold standard for automated review responses. Our baseline model leverages the popular

OpenNMT framework [11], which is originally based on Seq2Seq network that uses a bidirectional LSTM Encoder and a Unidirectional LSTM Decoder with attention. The encoder portion takes in embedding vectors of the source sentence and the decoder is trained on embedding vectors from labeled target sentence. The embeddings in the model are randomly initialized and are also learned during training. Multiplicative attention is applied for the decoder at each time step to determine which part of the encoded source sentence it should focus on. Subsequently, the decoder output at each time step is concatenated with the attention output which is then input to a softmax function to generate a probability distribution for the next word in the output sequence. For our purpose, we regard the customer reviews as sources sequence and the management responses as target sequences. In addition, we use the standard cross entropy loss.

3.3 Our Model

We design the following pipeline to specifically circumvent the misgenerated sentiment problem as noted in the previous sections. It comprises of a trigger module to activate a specific language generator.

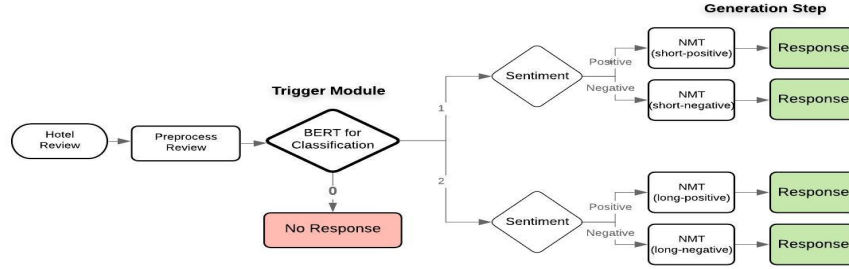


Figure 1: Life of a response. The figure presents the overview of inference

The trigger module is a fine-tuned BERT ternary sequence classifier. Our BERT model is built based on the pytorch-pretrained-BERT repository possessed by huggingface (<https://github.com/huggingface/transformers>). If the classifier returns 0, it means the incoming review is not worth responding; If the classifier returns 1 and 2, then it means the incoming review is worth a short and long response respectively. We manually set the threshold as 12 to distinguish short and long review responses. This is a hyper-parameter can be better fine-tuned given more time. Only when the BERT classifier returns 1 or 2, the subsequent models will be triggered. Otherwise, the pipeline stops here and our system simply returns nothing.

If the trigger model returns non-zero, we combine its output with the reviews' star rating to decide which Seq-to-Seq model we will trigger.

	1 or 2 stars	3 stars	4 or 5 stars
Short Response	short-negative	depends on the sentiment of the review	short-positive
Long Response	long-negative	depends on the sentiment of the review	long-positive

For 3 stars, it is a little bit more complicated. Since 3 star reviews are usually more neutral, we will feed a 3 star review review into a binary sentiment analysis model to predict whether it is positive or negative. We use the TextBlob module to detect the sentiment of a 3 star review. If the sentiment is found to positive, we pass the respective review into the positive language generator. Likewise, if the sentiment is negative, we pass it on to the negative language generator.

That being said, we trained 4 Seq-to-Seq models to generate responses. They are **short-negative**, **short-positive**, **long-negative** and **long positive** respectively. There are two reasons we did this:

- When we trained long and short responses in a single model. The model tended to be lazy and just return short responses for all test reviews. This is explainable because the short sequence can more easily lead to large likelihood than the long sequence so the model just favors more on short sequence. To mitigate this problem, we train long and short responses separately, which helps the semantic richness of the generated responses. We mainly report

the outputs of long models as the short models almost completely predict the phrase “Thank you”.

- If we train everything in a single model, during error analysis, we found it is hard for the model to get the sentiment of the review correct. This is the misgenerated sentiment problem. However, the review stars can be a very useful information to indicate the type of response desired. This is because, in most cases, low stars correspond to a negative review sentiment while high stars correspond to a positive sentiment. By training low and high star reviews on different models, we aim to improve the model’s ability of analyzing sentiments.

The Trigger module is inspired by Google’s Smart Reply Paper [4]. However, different from their paper’s trigger module which only returns "have response" or "no response", we came up with a ternary trigger module that classifies between "no response", "short response" and "long response". Additionally, we introduced a novel approach to train language generation models based on different sentiments, where the sentiment of a hotel review is indicated by the review’s star rating due to the uniqueness of the hotel review dataset. The Seq-to-Seq model uses the PyTorch version of open NMT [12]. The Bert sequence classifier refers to the "BERT Fine-tuning Tutorial with PyTorch" blog [13]. However, we significantly refactored and rewrote their code.

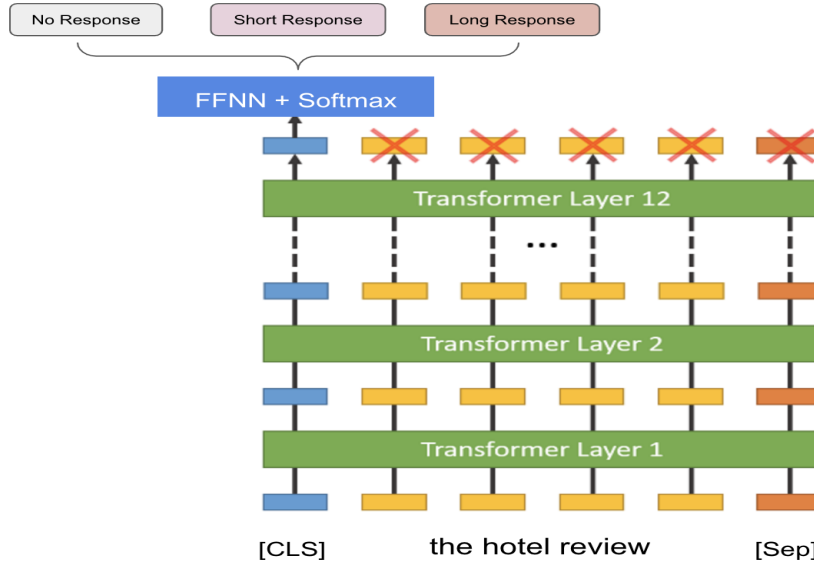


Figure 2: BERT Architecture

Here we go into further detail on the trigger module. The trigger module is a fine-tuned BERT classifier. The BERT model consists of 12 stacked bi-directional encoder transformers, each containing a multi-head self-attention layer and a feed-forward layer [Devlin et al., 2018]. The input hotel review is tokenized before feeding into the transformer and two special tokens: start [CLS] and end [Sep] are prepended and appended into the input sequence respectively. This BERT model already has pre-trained embeddings generated from masked language modeling, so we simply leveraged these embeddings to encode our input hotel review. A full connected feed forward neural network is concatenated after the BERT encoder and a softmax layer is applied to compute the probability of 3 different classes: no response, short response and long response.

On the output of the final (12th) transformer, only the first embedding vector is used for the classification task and all the other embeddings are discarded. As quoted from the original BERT paper, the first token of every sequence is always a special classification token ([CLS]). The final hidden state corresponding to this token is used as the aggregate sequence representation for classification tasks.

4 Experiments

4.1 Data

We use scraped TripAdvisor data from 2001-2016 from 144452 hotels. We have scraped the review, the star rating, and the respective management response if it exists.

We use different amounts of data for the tasks as our project progressed and we manually checked more and more data. We use 300,000 random reviews from TripAdvisor for the trigger model. 1/3 of these reviews had long responses (greater than 12 words), 1/3 had short responses, and 1/3 had no responses. 80% were used for training and 20% were used for testing. For the positive and negative models, 900,000 were used with an 80-20 training split.

4.2 Evaluation method

For our trigger module, we report accuracy and the confusion matrix. For our language model, we report the F1 scores from Rouge-1 and Rouge-2. We also report the response length of the generated models. In the appendix, we compare the distribution of predicted responses' sentiment and subjectivity with the actual responses'.

4.3 Experimental details

For all of our models, we used the Adam optimizer to train. The baseline seq2seq had a learning rate of 0.001. We used the default and recommended training parameters from OpenNMT: learning rate of 0.001, 100000 training steps, and a batch size of 32. While for the trigger module, we use the 12-layer BERT base model with a batch size of 4, a learning rate of 0.00002 and 4 epochs. The maximum length of the training data for the trigger module is set as 256 to avoid a token length that is greater than 512 because the "bert-base" model we use can only accept maximum 512 token size. For a review that is longer than 256 words, we only feed its first 256 words into the model, additional words are discarded. The model takes 17 hours to complete training on a Tesla M60 GPU.

4.4 Results

4.4.1 Trigger module results

We report the confusion matrix for the BERT trigger module's test accuracy.

	No Response	Short Response	Long Response
No Response	45.24%	23.66%	31.11%
Short Response	26.75%	47.35%	25.90%
Long Response	30.81%	23.06%	46.12%

4.4.2 Language Generation Highlight Results

- **Negative Review-**

- **Review:** *Dirty bathrooms. I had to wipe floor clean before stepping on it barefoot. It left the towel filthy and Towel was hairy...EW. Beds were clean. A plus. They were also conducive to sleep, another plus. Unprepared for large check-ins.*
- **Custom Pipeline Predicted Response:** *Thank you for taking the time to review your recent stay. I am very sorry to hear that your stay was not up to par. Please know that your concerns have been addressed with our housekeeping department. I hope you will give us another chance to prove how much we value your business.*
- **True Response:** *While I am pleased you enjoyed our Sweet Dreams bedding, I am disappointed to read of the lack of attention to detail in the cleanliness of your room. I am very sorry we let you down and will share your comments with our housekeeping management. Sincerely, «Name» General Manager*

- **Positive Review-**

- **Review:** *We go in kinda late and didn't spend a lot of time here. But,. the room was clean and fresh. The bed was great. The staff was very good. We got up after the bistro*

closed but were treated promptly to coffee when we asked. Would definitely stay here again.

- **Custom Pipeline Predicted Response:** *Thank you so much for taking the time to share your experience with us. We are so happy to hear that you enjoyed your stay with us and we look forward to your next visit!*
- **True Response:** *Thank you for your kind comments. I am delighted you enjoyed your stay, we look forward to seeing you back real soon!Sincerely,*

4.4.3 Comparing Baseline with Custom Pipeline

In Table 1 below, we calculate the number of “Thank you” reviews mispredicted. Specifically, we count the number of responses that are simply “Thank you” when the ground truth response does not contain the phrase “Thank you”. This is an estimation of how prevalent the misgenerated sentiment problem is. Our custom pipeline outperforms the baseline by an order of magnitude. Additionally, in Table 2, we show that the average length of our generated responses is longer than the baseline’s predictions, mitigating the problem of generating short, general responses.

Model	Performance
Baseline Model	0.0005949970250148749
Custom Pipeline	1.6403167998512778e-05

Table 1: Proportion of responses that are simply **“Thank you”** when the ground truth response does not contain any substring “Thank you” (Checking for punctuation and cases). Custom Pipeline combines both negative and positive sets. Only using the long models.

	Baseline	Custom Pipeline	Actual
mean	28.793066	32.444269	60.16426
std	10.423735	9.073518	24.74704
min	0	1	1
25%	23	26	42
50%	29	32	60
75%	36	39	82
max	100	100	7411

Table 2: Lengths of Predicted and Actual Responses on the long models. Custom Pipeline combines both negative and positive sets. (Note we took a larger sample for actual as it didn’t require the language generation preprocessing to calculate the length)

We present the distribution of F1 Scores from Rouge-1 and Rouge-2 across the test sets in Figure 3. Each score is calculated at the response level. The mean Rouge-1 F1 Score for the baseline is **0.314**, for the positive model **0.338**, and for the negative model **0.280**. The mean Rouge-2 F1 Score for the baseline is **0.107**, for the positive model **0.121**, and for the negative model **0.064**. There is not too much differentiation in these results, which is to be expected given the complexity of generated language. We would expect similar overlap of common words on ngrams, on average. What is interesting is that the negative model has the lowest performance in this metric. We hypothesize that this might be because negative review responses are more complex than positive reviews. The negative review responses might try to address specific context in the respective review, rather than just expressing gratitude, like a positive review response might.

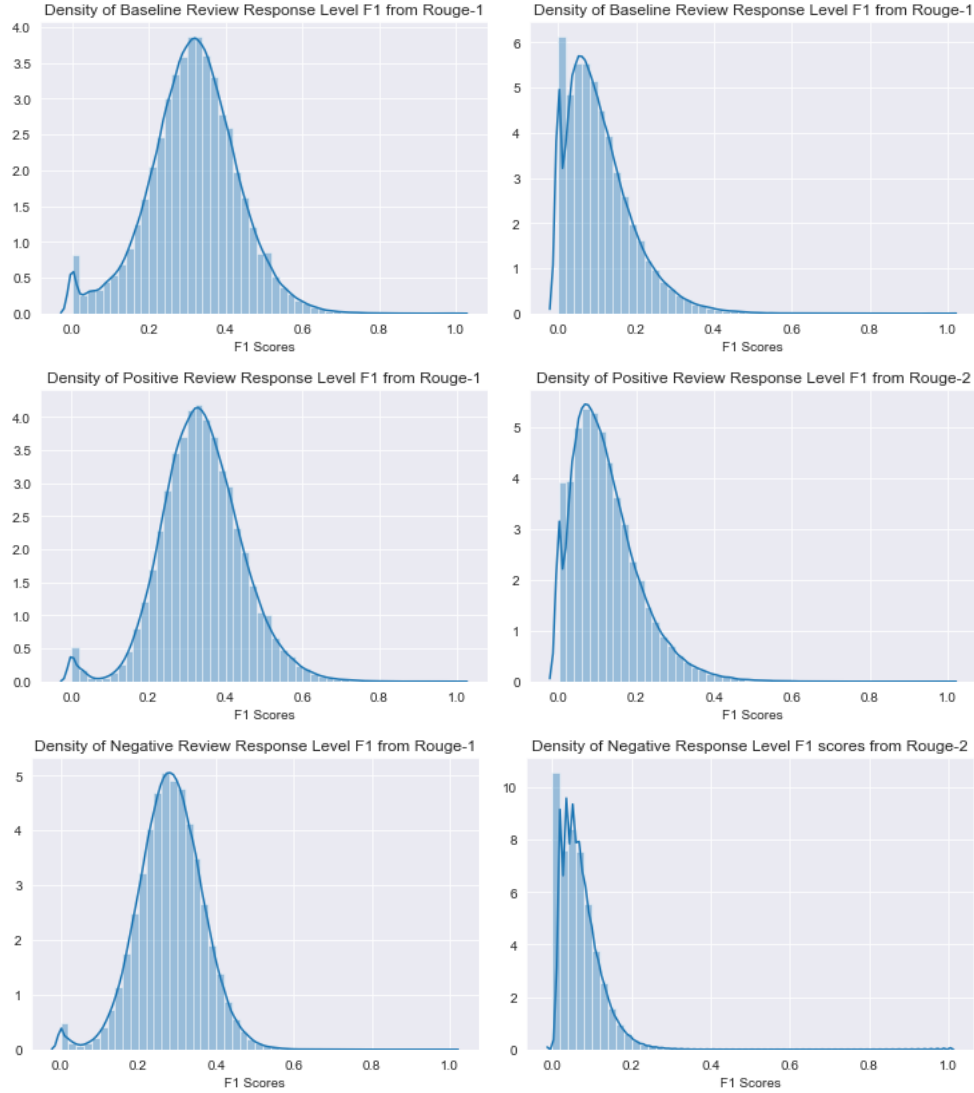


Figure 3: Distribution of F1 Scores from Rouge-1 (Left Column) and Rouge-2 (Right Column) for long models. The baseline model is on the first row, positive model is on the second, and the negative model is on the bottom row.

5 Analysis

5.1 Trigger Module

The Bert classifier on average delivers a 46% accuracy on predicting the right classification. We analyzed our data and believe there are two reasons why the classifier performs sub-optimally on this task:

- The problem itself is hard, For example: the review "It was a wonderful experience from the moment we walked through the front door. Laura was so helpful and friendly. She helped us with our rooms and recommended some great places to visit. Our rooms were spacious, comfortable and spotless." has a response "Happy to hear you enjoyed your stay and the friendly staff. Look forward to serving you in the future.", which happens to be a long response. However, short responses like "Happy you enjoyed your stay, hope see you soon!" can also be very reasonable. It is sometimes quite flexible whether we should output short or long responses to the given review.

- We plot the histogram of different response lengths, there is no clear boundary between short and long responses. Additionally, we manually pick the cutoff length between short and long responses as 12. A more fine-tuned cutoff length could lead to better results.

5.2 Seq-to-Seq

5.2.1 Human Analysis

Per [14], Human judgments are regarded as the gold standard for natural language generation task, we did human analysis on 25 randomly picked responses generated by our model (12 positive responses, 13 negative responses) in terms of Relevance, Sentiment, Fluency and Richness.

	Relevance	Sentiment	Fluency	Richness
Score	82%	92.0%	88%	36%

The human analysis results show that our model performs best on the sentiment analysis of the review, thanks to the training per different review stars. The model also has a good performance on relevance and fluency, which means our model can pick up the context of the review and generates logical and grammatically meaningful responses fairly well. The biggest shortcoming of our model is the richness of the response. The model still prefers to output more generic responses like "Thank you for taking the time to review our property. I am very sorry that you were not happy with your accommodations." instead of suggesting more specific responses. However, lacking of richness is a common problem in natural language generation. Google's Auto Reply paper [4], which also used a bidirectional LSTM, only generates short email responses. Algorithms like "directly upweight rare words during beam search" or "use a sampling decoding algorithm" are reported to be helpful to improve the richness. These will be the next steps to future improve the results of this paper.

6 Conclusion

Responding to customer reviews on online platforms can help hotels create a better customer experience and improve their ratings. A plethora of reviews received by hotels on such websites each day makes it difficult for them to manually address the reviews, and consequently, automating the review response process is extremely useful for hotel managers. In this work, we proposed a novel approach that incorporates review sentiments and context into the NMT model. Our analysis using automated metric as well as human evaluation illustrate our proposed model that uses a custom pipeline outperforms the baseline approach that only relies on the basic NMT model. In future, we can refine and expand our work by evaluating on a even larger dataset and finally deploying the model with industry partners in hotel business.

7 Appendix (optional)

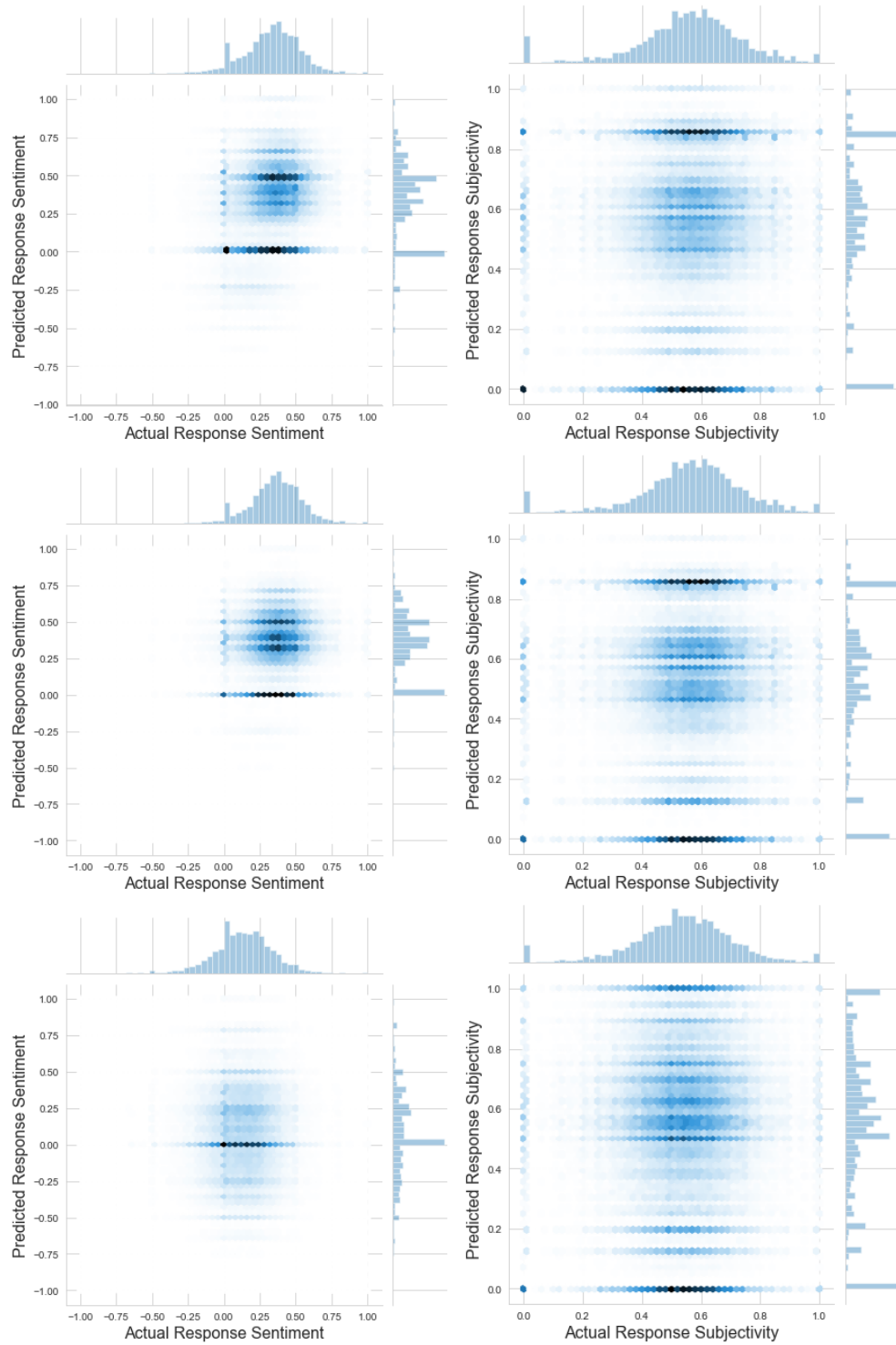


Figure 4: **1st Column:** Plots of Predicted Response Sentiment against the Actual Response sentiment. **2nd Column:** Predicted Response Subjectivity against the Actual Response sentiment. The baseline model is on the first row, positive model is on the second, and the negative model is on the bottom row. The sentiments and subjectivities are calculated using the TextBlob Module

References

- [1] Michael Luca. Reviews, Reputation, and Revenue: The Case of Yelp.Com. *SSRN Electronic Journal*, 2012.
- [2] Paul Phillips, Stuart Barnes, Krystin Zigan, and Roland Schegg. Understanding the Impact of Online Reviews on Hotel Performance: An Empirical Analysis. *Journal of Travel Research*, 56(2):235–249, 2017.
- [3] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian Yun Nie, Jianfeng Gao, and Bill Dolan. A neural network approach to context-sensitive generation of conversational responses. *NAACL HLT 2015 - 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, pages 196–205, 2015.
- [4] Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufmann, Andrew Tomkins, Balint Miklos, Greg Corrado, László Lukács, Marina Ganea, Peter Young, and Vivek Ramavajjala. Smart reply: Automated response suggestion for email. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 13-17-Aug:955–964, 2016.
- [5] Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-hsuan Sung, Laszlo Lukacs, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. Efficient Natural Language Response Suggestion for Smart Reply. 2017.
- [6] Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. The Design and Implementation of XiaoIce, an Empathetic Social Chatbot. *Computational Linguistics*, (May):1–62, 2020.
- [7] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 4(January):3104–3112, 2014.
- [8] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.
- [9] Building smart replies for member messages. <https://engineering.linkedin.com/blog/2017/10/buildingsmart-replies-for-member-messages>. Accessed: 2020/03/10.
- [10] Siyuan Jiang and Collin McMillan. Towards Automatic Generation of Short Summaries of Commits. *IEEE International Conference on Program Comprehension*, pages 320–323, 2017.
- [11] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. OpenNMT: Open-source toolkit for neural machine translation. *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of System Demonstrations*, pages 67–72, 2017.
- [12] <https://github.com/OpenNMT/OpenNMT-py>.
- [13] Fine tuning bert. <https://mccormickml.com/2019/07/22/BERT-fine-tuning/>.
- [14] <http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture15-nlg.pdf>.