Politeness Style Transfer on Conversational Data

By Yashmeet Gambhir and Barrett Lattimer

I. Problem Description

In recent years, state of the art language modeling has led to near human performance on many natural language processing tasks and can be used to create extremely advanced conversational experiences. A major drawback is that large AI systems trained on Wikipedia corpuses and large web crawl data sets generate language that contains only one specific, generic style of writing found on the internet. In a conversational setting, being able to generate appropriate emotions or respect to a user would be vital in create a truly empathetic and user-friendly conversational experience in various industries such as education, mental health therapy, online reservations/booking, tutoring, and home assistance. We also see a critical use case in applying politeness transfer to translation products to encourage respectful speech in new language learners. For our project, we wish to focus on a relatively new task of politeness transfer. The input will be a sample of conversational text, and the output will be a translated text in a polite tone. For example, the phrase *Send me the data* could be translated to *Can you please send me the data?.* The phrase *Say that again* could be translated to *Pardon me?* Our model could then be applied to conversational or translation agents by creating an extra politeness processing step to create a polite, desired output.

II. Related Work

Politeness - A significant contribution politeness of natural language for model training was brought by Danescu-Niculescu-Mizil et al.[1], who developed a computational approach to measuring politeness in text and used this approach to create a labeled dataset of over 10,000 utterances from collections of Wikipedia Talk and Stack exchange. The state of art and foundational paper for politeness transfer was developed by Madaan et al[2]. This team formally introduces the task of politeness transfer, and use a 'tag and generate' approach to selecting words to replace in the source sentence with politeness while applying it to a corporate email dataset. Other work in politeness text transfer focuses on an LM generating polite text or incorporating politeness into neural machine translation (Niu and Bansal[3], Sennrish et al.[4])/

Style Transfer/generation - Style transfer in text has taken many forms, and every machine learning approach (unsupervised, supervised, semi-supervised, and reinforcement learning) has been utilized. A recent, state of the art approach in sentiment and topic controlled text generation was produced by Dathathri et al[4]., who utilized pre-trained Transformer models and simple, bag of words style discriminators to generate text of a given emotion or topic. Jahmtani et al.[5] train a seq2seq model with a copy mechanism to transfer English to Shakespearean language using an available parallel corpus. A more common setting is when there is no parallel dataset available. Reinforcement learning can be applied with generative models and adversarially trained discriminators to create reward functions based on style classification score and fluency[6]. The use pre-trained classifiers can also be used to train special 'style' embeddings, as shown by Fu et al.[7]

Large pre-trained Transformers - In recent years, large pre-trained transformer models like BERT (Devlin et al.[9]) and GPT-2 (Radford et al.[10]) have been used to achieve state of art results in many language generation and classical NLP tasks. GPT-2 is often favored for language generation tasks due to its autoregressive training object and massive scale. However, Rothe et al. show the efficacy of using BERT on sequence to sequence tasks such as machine

translation, text summarization, and sentence fusion. Since GPT-2 has more powerful prior language generation power, we prefer its use for our experimentation.

Our approach is novel because we will be utilizing state of art methods in style transfer in a new domain of politeness, and our use of large conversational datasets should produce model with better application to conversational and translation settings.

III. Methods

Dataset - We choose to pull from a large scale conversational dialogue dataset *OpenSubtitiles*[8] which contains over 160 million dialogue sentences translated in various languages. Although it is traditionally used for machine translation, the English utterances provide more than enough data to represent this conversational domain. We choose a subset of 1 million utterances (from the beginning of the whole directory) as our initial dataset. We eliminate any inputs with single words and inputs with profanity to prevent profane generation[12]. A set of special characters (~/...) are also removed to simplify generation. It is more impactful to be able to translate rude or standoffish tones into politer tones, thus we further process this dataset to identify these. We a binary politeness classifier provided by the Cornell Convo-kit tool[7] (details discussed later) and select those utterances with scores below .10 (0-1 continuous scale). We experiment with various samples from this new dataset of ~26,000 rude utterances.

Model Design - Because we're working in an unsupervised setting, we must design a generation loss function that subjectively encourages the model to generate polite text that is both fluent and a close translation of the original input. Similar to Gong et al.[6], decide to use a reinforcement learning approach that rewards the generation of polite, fluent, and similar outputs. The decoder can be seen as an agent, its model parameters as the 'policy' it attempts to learn, and generated words as the action space. As seen in Figure 1., we use a pre-trained Transformer model to generate a sequence (using top-k sampling at each step) for each input. This generated sequence gets passed to our reward module, which returns a scalar reward to calculate the reinforcement learning loss. This is used to calculate a reinforcement learning loss value, which propagates back through the Transformer network to update.

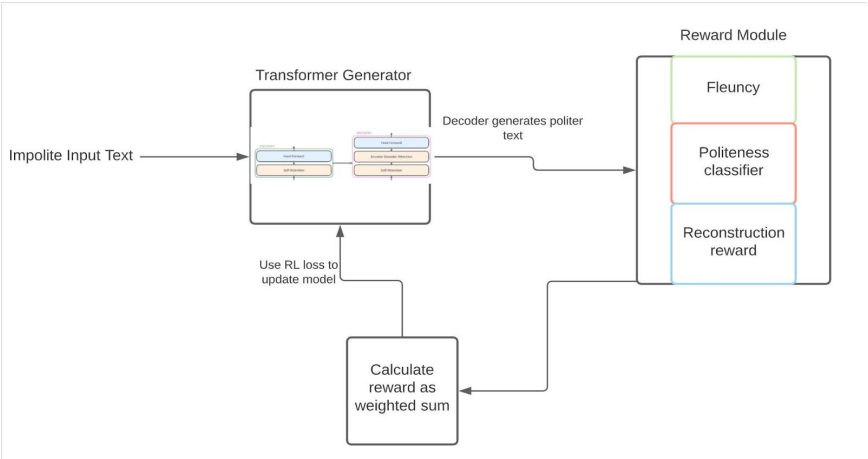


Figure 1.

Reward Module - The three rewards are responsible for pushing the model to create sentences that are fluent, polite, and similar to the input. To measure fluency, we use GPT-2 to calculate the perplexity of the generated sentences. Although this is the same as the generator we use, we hope that this would regularize the input to be as fluent as the GPT-2 output before we further train. However, a potential downside is that we favor our generator to maintain the same domain of generation (online written text instead of our new conversational domain). For politeness, we utilize the Cornell Convo-kit[11] classifier that achieves an accuracy of 76% trained and tested on a Wikipedia talk request dataset. The reconstruction reward compares the generated input to the output, and should be higher if the output is semantically similar to the input. For this reward we use the TF-IDFVectorizer object of the sklearn library, that calculates word similarity using the TF-IDF features of each sentence. This reward provides a good heuristic for fast and efficient training, however a more semantic understanding module (such as BertScore or text entailment) could be used to further improve this reconstruction reward. The final reward can be written as a weighted sum of each individual reward:

$$R = \alpha * F_{politeness}(Y^s) + \beta * F_{similarity}(Y^s) - \gamma * F_{perplexity}(Y^s)$$

Loss - We experiment with a couple different loss functions, the first of which utilizes the above reward in its entirety. This RL loss follows the policy gradient method first formulated by (Williams et al.[13]) and applied to a text generation setting. If y_1^s , y_2^s , to y_L^s is the sampled sequence for one input to the generator, the expected reward and loss term are as follows.

$$J(G_\theta) = R \sum_{t=1}^{L} \log p(y_t^s | y_1^s ... y_{t-1}^2, x)$$

$$L_{rl} = -J(G_\theta)$$

In practice, we notice that the implementation of this loss leads to quick divergence of the model to a sub optimal minimum. We further experiment with a simpler loss function that takes into account the traditional cross-entropy loss and scales it by our politeness reward:

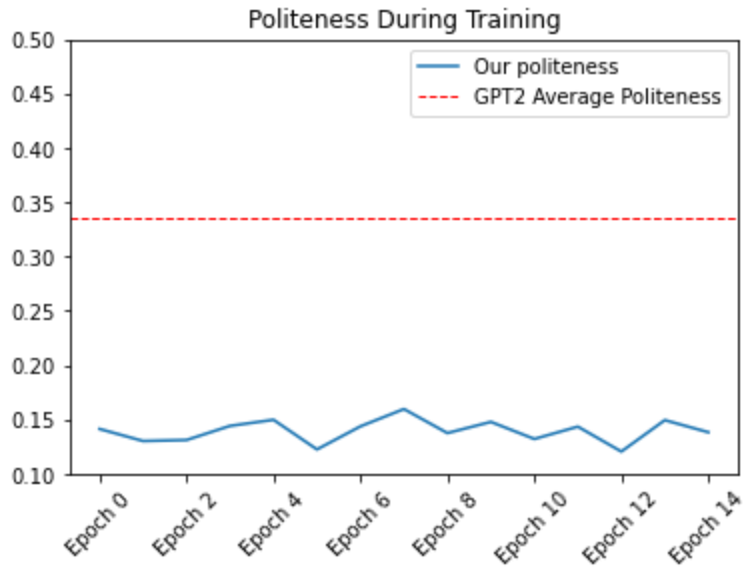
$$L = \frac{1}{F_{politeness}(Y^s)} * L_{ML}$$

Using this loss, we hope our model can learn the fluent, similar style and similar content output sentences. Note that the reward here is inverse, since we want to preserve the inverse relationship between politeness classification and loss for our model.

IV. Experiments and Results

Using our custom loss function that takes into account scaled politeness scores, we trained our model for 15 epochs. We optimized using Adam with a learning rate of 5*10^-5. We used a train test split on our dataset of 95% and 5%. Our dataset was the bottom 10% of rude subtitles that we found in the OpenSubtitles database. We used the original GPT2 model that we fine tuned as our baseline model and it is what we compared all of our results against. The experiments were not successful in increasing the politeness of generated sentences.

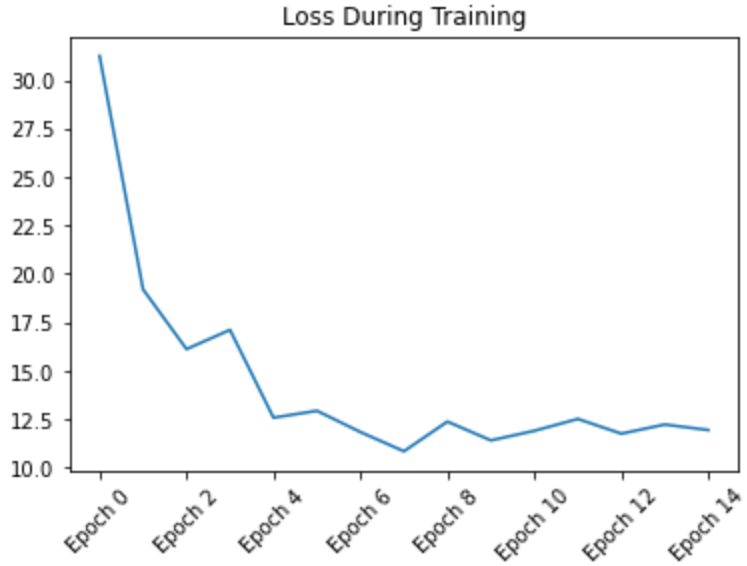
Examples:
Impolite:
Input: you can't refute facts. ----> Output: "and die" is the truth. " Politeness score .03
Input: you're quite a joker. -----> Output: "What? You're a liar." Politeness score .037
Polite:
Input: and i smoke a pipe. -----> Output:"I'm not a smoker. " Politeness score: .4
Input: and my hair is a mess. ----> Output: " I'm not sure what to do" Politeness score: .5



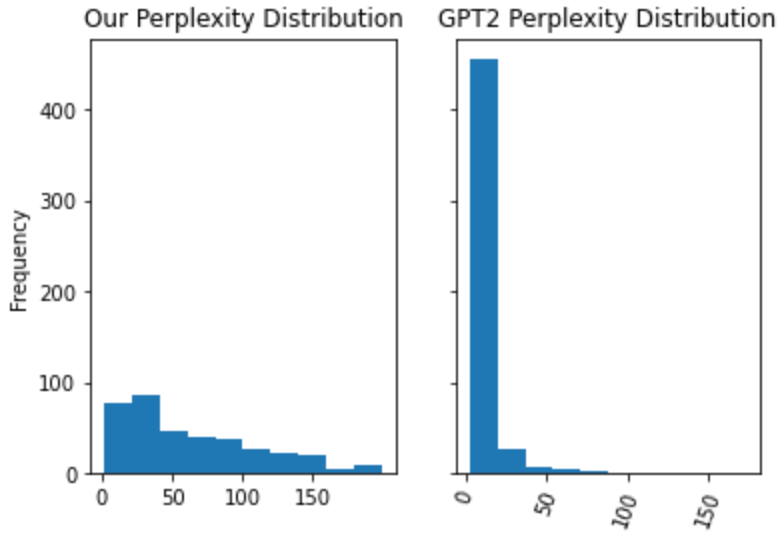
Above, you can see a few examples of the output that our model generates after training for 15 epochs on our reinforcement learning. You can also see that while our model did produce good scoring sentences in some instances, most of the time our model failed to reach average politeness for GPT2. We believe this phenomenon is due to the nature of our perplexity oriented loss function and the gradients focus on reinforcing that good perplexity.

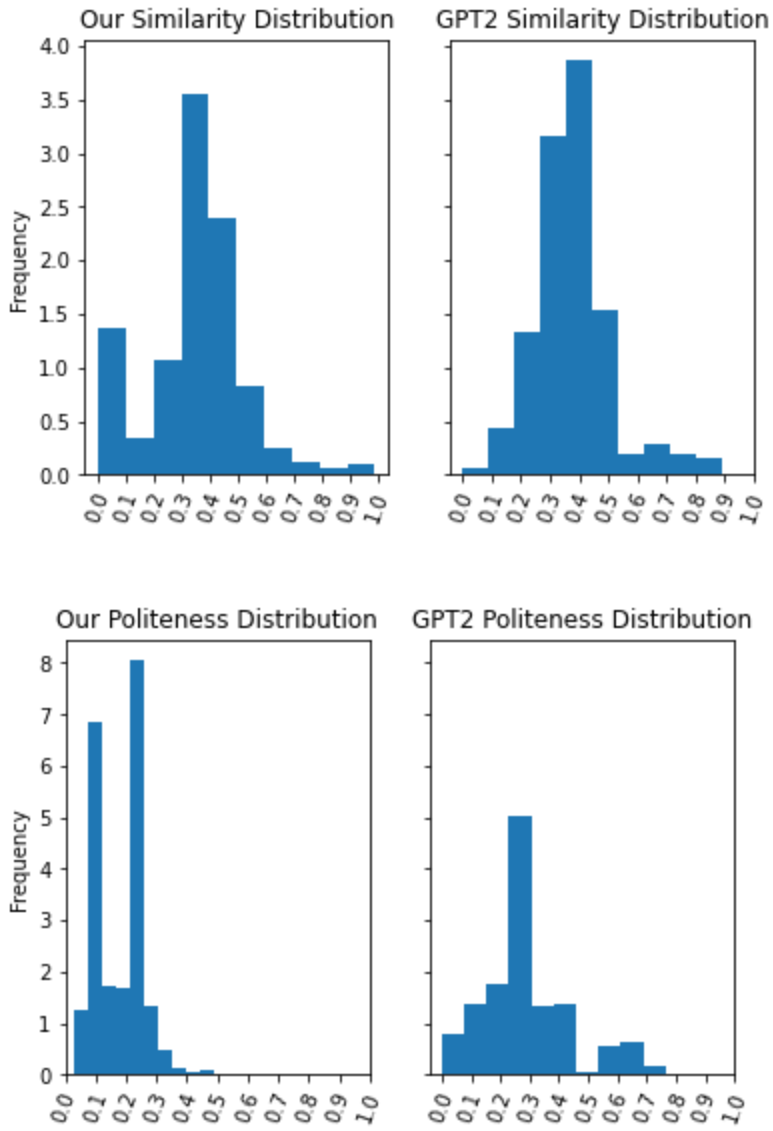
One important modification we had to make when running our experiments was limiting the size

of the generated output. We made this choice because our model began generating sentences that were the same length as the input and then repeating that sentence until max length. Although this did not affect the perplexity score, this issue severely impacted the politeness and similarity score, skewing our training for the worse. Limiting our generation size dynamically based on the size of the sentence helped us score more accurate similarity scores as well as give the model a clearer view of the politeness score.



The loss of our model did significantly decrease down to about 11 which is similar to the level of normal cross entropy loss that GPT2 achieves. Surprisingly, even though our model was trained strictly on politeness and essentially perplexity, our model showed a worse distribution for both perplexity and politeness scores. Similarity scores, however, showed promise with both models achieving essentially the same distribution. We believe that this phenomenon occurs because our model did not fully learn how to reproduce text rather than generating which is what the GPT2 head we used was originally trained to do.





V. Future Works

Our work has raised a few questions and given us the necessary skills to do a deeper dive into our politeness translation task. We would like to achieve higher average politeness than the pretrained GPT2. In order to achieve this goal, we need to first further modify our loss function to better incorporate politeness as a priority and not just perplexity. One potential benefit may be to look into other pretrained models that are more focused on translation rather than generation. Using a translation model would prevent us from having to train our model to be polite and also reproduce rather than generate. On the same note, a method we did not get to try was first training GPT2 to reproduce a sentence whether it was polite or not, take a benchmark, and then use reinforcement learning to blend in the importance of politeness on the new benchmark. Doing this two stage training could potentially simplify the problem and be easier to model in each case rather than together. Lastly we want to try incorporating freezing layers into our

model training. We could possibly train our model to reproduce text and then only train the final layer on instilling politeness on the reproduced sentence. Using freezing could help prevent the model from losing the importance of reproducing but also making some last minute tweaks to improve politeness.

VI. Conclusion

The problem of style transfer and instilling a certain feeling or politeness on a text is a recent area of interest in the NLP community. Even though there is significant interest in the topic, style translation specifically for politeness has no existing parallel datasets, begging the need for reinforcement learning. In this paper we used a custom loss function to tweak a pretrained GPT2 model to, one, reproduce sentences rather than generate follow up sentences, and two, instill politeness on this newly generated sentence. While our model did fail to reproduce the politeness scores of GPT2, it did learn with a custom loss function and still produce some very intelligent outputs.

References

(1) DANESCU-NICULESCU-MIZIL, C., SUDHOF, M., JURAFSKY, D.,LESKOVEC, J.,ANDPOTTS, C. A computational approach to politenesswith application to social factors. InProceedings of the 51st AnnualMeeting of the Association for Computational Linguistics (Volume 1:Long Papers)(Sofia, Bulgaria, Aug. 2013), Association for Computa-tional Linguistics, pp. 250–259

(2) MADAAN, A., SETLUR, A., PAREKH, T., POCZOS, B., NEUBIG, G.,YANG, Y., SALAKHUTDINOV, R., BLACK, A. W.,ANDPRABHUMOYE,S. Politeness transfer: A tag and generate approach. InProceedingsof the 58th Annual Meeting of the Association for Computational Lin-guistics(Online, July 2020), Association for Computational Linguistics,pp. 1869–1881

(3) NIU, T.,AND BANSAL, M. Polite dialogue generation without paralleldata.Transactions of the Association for Computational Linguistics 6(2018), 373–389

(4) SENNRICH, R., HADDOW, B.,ANDBIRCH, A. Controlling politenessin neural machine translation via side constraints. InProceedings of the2016 Conference of the North American Chapter of the Associationfor Computational Linguistics: Human Language Technologies(SanDiego, California, June 2016), Association for Computational Linguis-tics, pp. 35–40

(5) Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017a. Shakespearizing modern language using copy-enriched sequence-to-sequence models. EMNLP 2017, 6:10

(6) GONG, H., BHAT, S., WU, L., XIONG, J.,ANDHWU, W.-M. Reinforce-ment learning based text style transfer without parallel training corpus.InProceedings of the 2019 Conference of the North American Chapter ofthe Association for Computational Linguistics: Human Language Tech-nologies, Volume 1 (Long and Short Papers)(Minneapolis, Minnesota,June 2019), Association for Computational Linguistics, pp. 3168–3180

(7) Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pages 663–670

(8) http://opus.nlpl.eu/OpenSubtitles-v2018.php

(9) Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pretraining of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics

(10) Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pretraining. Technical report, OpenAI

(11) https://convokit.cornell.edu/

(12) https://pypi.org/project/better-profanity/

(13) Ronald J Williams. 1992. Simple statistical gradient following algorithms for connectionist reinforcement learning. In Reinforcement Learning, pages 5–32. Springer.