Proto-Gen: An end-to-end neural generator for persona and knowledge grounded response generation

Sougata Saha, Souvik Das, Rohini Srihari
State University of New York at Buffalo
Department of Computer Science and Engineering
{sougatas, souvikda, rohini}@buffalo.edu

Abstract

In this paper we detail the implementation of Proto-Gen, an end-to-end neural response generator capable of selecting appropriate persona and fact sentences from available options, and generating persona and fact grounded responses. Incorporating a novel interaction layer in an encoder-decoder architecture, Proto-Gen facilitates learning dependencies between facts, persona and the context, and outperforms existing baselines on the FoCus dataset for both the sub-tasks of persona and fact selection, and response generation. We further fine tune Proto-Gen's hyperparameters, and share our results and findings.

1 Introduction

With the growth of neural methods for language modelling, the task of response generation in the field of open domain dialogue and interactive systems have witnessed significant improvements. Incorporating transformer (Vaswani et al., 2017) based architectures with billions of parameters, and trained on large training corpora, such models (Radford et al., 2019; Zhang et al., 2020; Roller et al., 2021; Xu et al., 2022) have advanced the state-of-the-art in response generation. However, trained with the objective of generating the next response by conditioning only on the context, such models often result in unnatural and hallucinated responses (Rashkin et al., 2021), which if not addressed appropriately, hampers it's usefulness in practical settings (Saha et al., 2021).

Although recent years have witnessed advancements in response generators which can factor in external knowledge (Dinan et al., 2019; Gopalakrishnan et al., 2019) and exhibit certain human-like features like personality traits, emotions, .etc (Mairesse and Walker, 2007; Zhang et al., 2018; Rashkin et al., 2019; Saha et al., 2022), research in response generators that can generate user-centric responses by factoring both user persona and external knowledge is still an unsolved problem. In this paper we propose Proto-Gen, an end-to-end response generator that can select the most appropriate fact and user persona sentences based on the conversation context, and generate a response customized for the user.

2 Task an Data Description

The task aims at engendering intelligent response generators that can generate appropriate response to user queries by factoring in the user's persona along with available external facts. It is further divided into two sub-tasks:

- Persona sentences and knowledge prediction: With the inputs being 5 persona candidates of the user, 10 knowledge candidates pertaining to the topic of discussion, and the conversation context, this sub-task requires predicting the correct persona and knowledge sentence which can be used for generating the response.
- Response generation: This sub-task requires generating the agent response to the user query in natural language, using persona and knowledge sentences.

The dataset (Jang et al., 2022) comprises 14,452 persona-knowledge dialogues (11,562 training, 1,445 validation, and 1,445 testing) pertaining to discussions about landmarks such as Statue of Liberty, Eiffel Tower, The Great Wall, etc.

3 Methods

As illustrated in Figure 1, we implement an end-to-end encoder-decoder based architecture for jointly performing all sub-tasks. Below we discuss each component in detail.

3.1 Encoding

The encoding layer comprises two BART (Lewis et al., 2020) based encoders: (i) **Query Encoder**

Proceedings of the 29th International Conference on Computational Linguistic, pages 9–14
October 12–17, 2022.

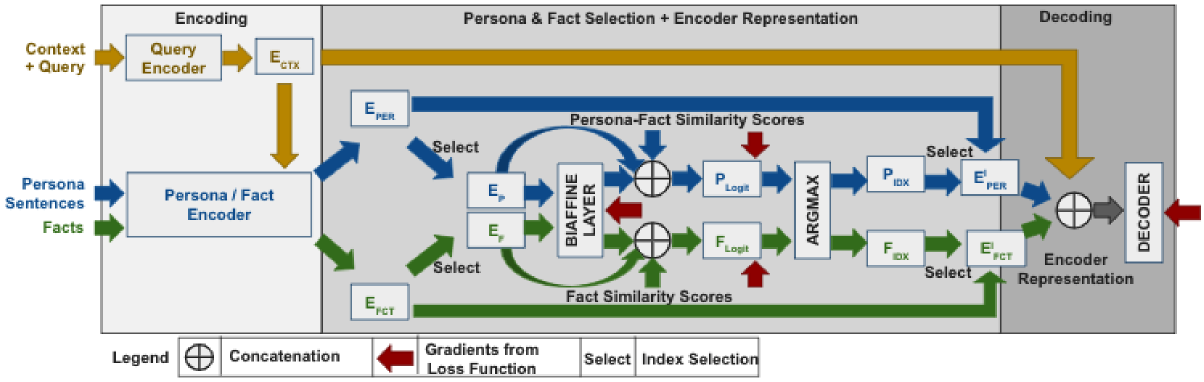

Figure 1: Proto-Gen End-to-End Model Architecture.

for encoding the conversation context and query. (ii) **Persona/Fact Encoder** for sequentially encoding the available persona and fact sentences. First the query encoder Q_Enc encodes the context CTX, which comprises the last 128 tokens of the concatenated previous turns and the current user query (Equation 1). The persona and fact encoder PF_Enc sequentially encodes each of the 5 persona and 10 knowledge sentences, which are further combined with the encoded context $\mathrm{E_{CTX}}$ using multi-headed attention MHA followed by dropout Drop (Equations 2 to 5), to yield the final persona and fact encodings $\mathrm{E_{PER}}$ and $\mathrm{E_{FCT}}$.

$$\mathrm{E_{CTX}} = \mathrm{Q\_Enc(CTX)} \tag{1}$$

$$\mathrm{E_{PER}} = \mathrm{PF\_Enc(P^i)}|_{i=1}^{5} \tag{2}$$

$$\mathrm{E_{FCT}} = \mathrm{PF\_Enc(F^i)}|_{i=1}^{10} \tag{3}$$

$$\mathrm{E_{PER}} = \mathrm{E^i_{PER}} + \mathrm{Drop(MHA(E^i_{PER}, E_{CTX}))}|_{i=1}^{5} \tag{4}$$

$$\mathrm{E_{FCT}} = \mathrm{E^i_{FCT}} + \mathrm{Drop(MHA(E^i_{FCT}, E_{CTX}))}|_{i=1}^{10} \tag{5}$$

3.2 Interaction Layer

The interaction layer captures interactions between the context and the presented persona and fact sentences, for determining the best suited persona and fact sentences for generating the current response. The layer inputs the encoded context $\mathrm{E_{CTX}}$, persona $\mathrm{E_{PER}}$ and fact sentences $\mathrm{E_{FCT}}$, and outputs a final concatenated representation $\mathrm{E_{ENC}}$ for the decoder.

For determining the most appropriate persona and fact sentences for the current turn's response, the interaction layer utilizes fully-connected neural networks (FNN) which input a concatenated representation of:

1. Biaffine Interaction Logits: The logits sc from a biaffine classifier which captures the interactions between the input persona and fact sentences. Biaffine classifiers are generalizations of linear classifiers, which include multiplicative interactions between two vectors (Dozat and Manning, 2016). Hence, we incorporate a biaffine layer for jointly determining the most appropriate persona and fact sentences for the current turn. Using layers of FNNs, the embedding of the start-of-sequence (SOS) token of both the fact and persona sentences are transformed to a reduced hidden size, which in turn are passed through a biaffine classifier to predict the most appropriate pair of persona and fact sentences for response generation (Equations 6 to 9). This layer is trained by minimizing the binary cross-entropy (BCE) loss between the predicted logits and the actual labels (Equation 16).

2. Persona & Fact Prior Logits: Depicted in Equations 10 and 11, FNNs are used to compute the prior probability of independently selecting each persona and fact sentence in the current turn. The FNNs inputs the representative persona and fact vectors $\mathrm{E_P}$ and $\mathrm{E_F}$ and yields the logits $\mathrm{FNN(E_P)}$ and $\mathrm{FNN(E_F)}$ for each sentence.

3. Pre-computed Similarity Vector: We input two additional vectors comprising normalized Levenshtein based similarity scores [1], which act as biases. (i) $\mathrm{F_{sim}}$: A vector comprising unit normalized similarity scores between each factual sentence and the available Wikipedia knowledge for the landmark of discussion. (ii) $\mathrm{P_{sim}}$: A vector comprising unit normalized similarity scores between the most similar fact from step (i), and the available persona sentences.

Equations 10 and 11 details the fact and persona prediction sub-tasks, which are trained by minimiz-

[1] https://pypi.org/project/fuzzywuzzy/

ing the BCE loss functions (Equations 18 and 19). Finally, the interaction layer engenders the final representation of the encoding step by concatenating the encoded context $\mathrm{E_{CTX}}$, and the encodings of the most likely persona and fact sentences (Equations 12 to 14).

$$\mathrm{Get(X, idx) = X[idx, :]} \quad (6)$$

$$\mathrm{E_P = Get(E_{PER}, 0);\ E_F = Get(E_{FCT}, 0)} \quad (7)$$

$$\mathrm{Biaf}(x, y) = x^T \mathrm{U} y + \mathrm{W}(x \oplus y) + b \quad (8)$$

$$\mathrm{sc = Biaf(FNN(E_P), FNN(E_F))} \quad (9)$$

$$\mathrm{P_{logit} = FNN(Cat(FNN(E_P), sc, P_{sim}))} \quad (10)$$

$$\mathrm{F_{logit} = FNN(Cat(FNN(E_F), sc, F_{sim}))} \quad (11)$$

$$\mathrm{E_{PER}^{idx} = Get(E_{PER}, argmax(P_{logit}))} \quad (12)$$

$$\mathrm{E_{FCT}^{idx} = Get(E_{FCT}, argmax(F_{logit}))} \quad (13)$$

$$\mathrm{E_{ENC} = Cat(E_{CTX}, E_{PER}^{idx}, E_{FCT}^{idx})} \quad (14)$$

3.3 Decoding and Loss Function

We reuse BART's decoder layers for decoding, where the concatenated representation $\mathrm{E_{ENC}}$ is input to the decoder for generating the final response $\mathrm{y_{pred}}$ (Equation 15). Depicted in Equation 20, we train the model end-to-end by minimizing the aggregated interpolated loss across all sub-tasks with interpolation factors α, β and γ_1/γ_2 for language modelling loss (Equation 17), persona-fact biaffine interaction prediction loss, and persona/fact selection loss respectively. In order to enhance response generation, we also add an extra penalty term δ with interpolation factor λ to the aggregated loss function, which is set to be proportional to the ratio of salient tokens that are missing from the generated response, with the salient tokens being the nouns, adjectives and verbs in the golden response, which are pre-computed using Spacy [2].

$$\mathrm{y_{pred} = Decoder(E_{ENC})} \quad (15)$$

$$\mathcal{L}^{\mathrm{biaf}} = \mathrm{BCE(y_{biaf}, sc)} \quad (16)$$

$$\mathcal{L}^{\mathrm{LM}} = \mathrm{CE(y_{act}, y_{pred})} \quad (17)$$

$$\mathcal{L}^{\mathrm{PER}} = \mathrm{BCE(P_{act}, P_{logit})} \quad (18)$$

$$\mathcal{L}^{\mathrm{FCT}} = \mathrm{BCE(F_{act}, F_{logit})} \quad (19)$$

$$\mathcal{L} = \alpha\mathcal{L}^{\mathrm{LM}} + \beta\mathcal{L}^{\mathrm{biaf}} + \gamma_1\mathcal{L}^{\mathrm{PER}} + \gamma_2\mathcal{L}^{\mathrm{FCT}} + \lambda\delta \quad (20)$$

[2]https://spacy.io/usage/linguistic-features

4 Experiments and Results

4.1 Experiment Setup

We use BART (Lewis et al., 2020) as the base encoder, and increase its embedding layer to accommodate two special tokens <agent_1>, <agent_2> to distinguish between speaker turns, and two tokens <persona>, <knowledge> to distinguish between persona and factual sentences. Four layers comprising four attention heads are used for multi-headed attention in the interaction layer. The hidden size of the FNNs in the biaffine layer is set to 600. All models are trained with a learning rate of 1e-5 for 15 epochs and optimised using AdamW (Loshchilov and Hutter, 2017), with early stopping if the validation loss doesn't reduce for 2 epochs. Further, a weight of 5.0 is applied to positive examples during computing binary cross entropy loss for the biaffine prediction. The interpolation factors $\alpha, \beta, \gamma_1, \gamma_2$ and λ are set to 0.6, 0.1, 0.1, 0.1, and 0.1 respectively by default.

4.2 Experiments

We experiment with different hyperparameter settings to engender multiple variants of the model. Specifically, we experiment with (i) Adding/removing the additional persona and fact similarity score vector as inputs in the interaction layer, (ii) Adding/removing the keyword based penalty term δ in the final model loss (Equation 20), (iii) Using both the base and large versions of pre-trained BART, (iv) Adding dropout with a probability of 0.1 for regularization post concatenating the biaffine interaction logits, persona & fact prior logits and the pre-computed similarity vector in the interaction layer, (v) Sharing the same base encoder for encoding fact and persona sentences, (vi) Different values of the interpolation factor. Table 1 lists all the different hyperparameter settings that we experiment with, along with the resultant model ids.

4.3 Results and Observations

We train and evaluate all the model variants on the standard training and evaluation splits of the FoCus (Jang et al., 2022) dataset. For persona and knowledge selection (sub-task 1), we report overall accuracy scores-Persona Accuracy and Knowledge Accuracy, as well as Average Grounding-an average of the two accuracy scores. For response generation (sub-task 2), we report SacreBLEU (Post, 2018), CharF++ (Popović, 2015) and ROUGE-L

Model ID	Similarity Scores	Keyword Penalty	Base Model	Add Dropout	Persona & Fact Shared Encoder	Interpolation Factors
1	yes	no	bart-base	yes	yes	0.7, 0.05, 0.15, 0.1, 0.0
2	yes	no	bart-base	yes	yes	0.6, 0.2, 0.1, 0.1, 0.0
3	yes	no	bart-base	yes	no	0.6, 0.2, 0.1, 0.1, 0.0
4	yes	no	bart-base	no	yes	0.6, 0.2, 0.1, 0.1, 0.0
5	yes	no	bart-large	no	yes	0.6, 0.2, 0.1, 0.1, 0.0
6	yes	yes	bart-base	no	yes	0.6, 0.1, 0.1, 0.1, 0.1
7	no	yes	bart-base	no	yes	0.6, 0.1, 0.1, 0.1, 0.1

Table 1: List of experiments with different hyperparameter settings

Model ID	Persona Accuracy	Knowledge Accuracy	Average Grounding	Sacre BLEU	Char F++	ROUGE L	Average Generation	Average Score
(Jang et al., 2022)*	86.86	65.06	75.96	10.87	27.90	30.99	23.26	49.61
1	77.26	32.49	54.87	8.58	28.08	21.81	19.49	37.18
2	86.38	80.36	83.37	18.91	40.07	38.03	32.34	57.85
3	86.16	74.24	80.20	18.19	40.10	36.27	31.52	55.86
4	85.02	**85.18**	**85.10**	**19.85**	**42.32**	**38.84**	**33.67**	**59.39**
5	**87.75**	68.72	78.23	18.35	39.68	38.14	32.06	55.14
6	84.00	83.09	83.54	19.28	41.74	38.14	33.05	58.30
7	85.35	79.42	82.39	19.39	41.90	38.00	33.10	57.74

Table 2: Results of the experiments from Table 1. The best score for each metric is highlighted in bold. * lists the best scores from the external baseline.

(Lin, 2004) scores, along with an aggregated metric of all the three metrics-Average Generation. We also report Average Score-an overall metric for both the sub-tasks by averaging the Average Grounding and Average Generation scores.

Table 2 shares the results of the experiments listed in Table 1. We make the following observations: (i) Comparing models 4 and 5, we observe that using bart-base as the base model generally outperforms bart-large, which we attribute to the smaller size of training data in comparison to the larger number of parameter updates requires to train the large model. (ii) Comparing models 6 and 7, we see that incorporating the persona and fact similarity scores as additional vectors mostly results in better scores. This intuitively makes sense, as the similarity vector acts as an additional bias term for the model, which facilitates learning. (iii) Comparing models 4 and 6, we observe that adding the keyword based penalty term to the loss function does not seem to help learning. (iv) In comparison to model 4, adding dropout to the concatenated representation of the interaction layer in model 2 does not yield better results. We reason that since the base architecture already includes multiple regularization constrains, adding additional dropout layers hinders learning, specially because the size of the training data is small compared to the pre-training data of BART. (v) Comparing models 2 and 3, we observe that sharing the base encoder for encoding both persona and fact sentences, results in better scores. We attribute this to the fewer parameter updates required for parameter sharing. (vi) Comparing models 1 and 2, we note that a higher interpolation factor for biaffine classifier yields better overall scores, in comparison to fact and persona selection. Overall, we observe that model 4, which uses bart-base as the base model, inputs the additional similarity vectors, shares encoder for encoding persona and fact, while not adding additional dropout and keyword penalty, yields best results on the validation set.

5 Conclusion

Here we detail Proto-Gen, an end-to-end neural response generator, that can not only select appropriate persona and fact sentences from available input options, but also generate persona and knowledge grounded responses. Incorporating a novel interaction layer which includes biaffine classifiers and trained on the FoCus dataset, Proto-Gen outperforms existing external baselines for all sub-tasks. We further perform experiments to fine tune Proto-Gen's hyperparameters, and report our results.

References

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of Wikipedia: Knowledge-powered conversational agents. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Timothy Dozat and Christopher D. Manning. 2016. Deep biaffine attention for neural dependency parsing.

Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tür. 2019. Topical-Chat: Towards Knowledge-Grounded Open-Domain Conversations. In *Proc. Interspeech 2019*, pages 1891–1895.

Yoonna Jang, Jungwoo Lim, Yuna Hur, Dongsuk Oh, Suhyune Son, Yeonsoo Lee, Donghoon Shin, Seungryong Kim, and Heuiseok Lim. 2022. Call for customized conversation: Customized conversation grounding persona and knowledge. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10803–10812.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization.

François Mairesse and Marilyn Walker. 2007. PERSONAGE: Personality generation for dialogue. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 496–503, Prague, Czech Republic. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Hannah Rashkin, David Reitter, Gaurav Singh Tomar, and Dipanjan Das. 2021. Increasing faithfulness in knowledge-grounded dialogue with controllable features. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 704–718, Online. Association for Computational Linguistics.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*.

Sougata Saha, Souvik Das, Elizabeth Soper, Erin Pacquetet, and Rohini K. Srihari. 2021. Proto: A neural cocktail for generating appealing conversations.

Sougata Saha, Souvik Das, and Rohini Srihari. 2022. Stylistic response generation by controlling personality traits and intent. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 197–211, Dublin, Ireland. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Jing Xu, Arthur Szlam, and Jason Weston. 2022. Beyond goldfish memory: Long-term open-domain conversation. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.