Designing a recipe generation system.

Ankit Vadehra
MSCI 641 / UWaterloo

Abstract

We focus on the task of conditional narrative generation by focusing on automatic recipe generation. The task deals with generating coherent and comprehensive recipe generation. Recipe title and ingredients are considered as the conditional parameters and the aim is to generate an accurate recipe that utilizes all the ingredients and is practical.

1 Introduction:

Conditional/controlled text generation is a widely studied problem in natural language processing. The approach has been studied for various different problems like dialogue generation, data-to-text generation, creative text generation like story, music, poem etc. Certain improvements like attention, textual hierarchy, variational latent representation have also been utilized to improve generation. We use this section to mention some recent work in the field of controlled text generation, recipe generation and general purpose text generation models.

Li Et al. (Li et al., 2016) designed a conversational reply generation system that utilized a conditional external embedding for each specific user. There are various other approaches that utilize a similar approach like the work by Xing Et al. (Xing et al., 2017) which focuses on prospective topics as conditional parameters. Approaches like these have been utilized in various different seq2seq problems like News Headline generation (Zhang et al., 2018), lyrics generation (Potash et al., 2015), story generation (Fan et al., 2018, 2019) where apart from the corresponding input as a conditional parameter an external latent representation is utilized for generation as well. Considering that there are too many papers in this domain we focus on the approaches to improve text generation.

One of the first approaches to improve text generation is the introduction of an attention mechanism in the Seq2Seq (encoder-decoder) model. The work by Bahdanau Et al. (Bahdanau et al., 2014) and Luong Et al. (Luong et al., 2015) proposed an attention mechanism where the encoder outputs are assigned weightage(attention) when given to the decoder for response generation. Using the attention weights the decoder can focus on certain specific words in the input at each time step of generation. Attention is a very useful in text generation. The subsequent work by Vaswani et al. (Vaswani et al., 2017) proposed the transformer architecture which eliminated the RNN component of the Seq2Seq model and showed that using attention can be very useful in generating text. This approach has been further used to design some of the most powerful and massive text generation transformer models like BERT (Devlin et al., 2018) and GPT2 (Radford et al.).

Many NLP tasks make use of long text documents. Documents spanning multiple sentences and paragraphs. Simple RNN encoder-decoders are not able to handle such long range texts. As a result a hierarchical model was proposed by Sordoni Et al. (Sordoni et al., 2015) which was used to generate sequential web query suggestions based on a users past searches. The hierarchical model makes use of an additional contextual encoder that makes use of the encoder outputs to maps and learn the long form/range dependence of the input text. A similar model was modified and used by Serban Et al. (Serban et al., 2015, 2016) to model and generate a multi-turn dialogue system. The encoder takes in the input and the contextual encoder can learn the current state of the conversation. The hierarchical component is helpful to learn and generate text efficiently while being able to also hold meaningful and longer turn sentence level text generation.

Another approach to improve text generation makes use of a variational latent representation.

These models sample from a stochastic latent representation and as a result can generate multi-variate or diverse response for the same input. This gives them a bit more flexibility while still being able to generate efficient responses. The variational model was adopted from the VAE model for images (Pu et al., 2016) and has been used to generate dialogues (Serban et al., 2017) as well as stylistic text with different controlled setting like sentiment, tense, length etc (Logeswaran et al., 2018; Kikuchi et al., 2016; Hu et al., 2017).

Another interesting approach is to use adversarial learning to generate stylized text. Sequential GAN's have been successful is generating text of this form (Li et al., 2017). Another approach is to make use of the discriminator used in GAN's to destylize a latent representation of its respective style and then using style specific controllable features like decoders, or style embedding to generate controlled text. Fu Et al. (Fu et al., 2018) used a Seq2Seq model of this type to generate style specific text.

There are various other approaches explored to generate such controlled text like editing networks, memory banks, data-to-text, knowledge graphs etc. We do not mention those approaches since they are built on a different approach.

2 Method

To tackle the problem of recipe generation we present three models that act as the baseline for future work on introducing style specific controlled latent variables. We focus on a Seq2Seq Model, a Seq2Seq+Attention Model and and HRED+attention hierarchical model.

We use these three approaches to compare prospective useful components for an efficient and comprehensive narrative text generation system. We focus on three aspects of generation - Reconstruction, Feasibility and Comprehensiveness. We explain these metrics in the experiment section.

We use this section to briefly explain the three models of our experiment.

2.1 Data

As a precursor all three of our models deal with 3 types of data. The title, ingredients and instructions for the recipe.

2.2 Seq2Seq Model

For the simplest baseline we design a Seq2Seq model that tries to generate the recipe instruction for a food item given the set of ingredients and titles as an input. Figure 1 refers to our baseline model.

In our seq2seq model we generate the appropriate representation for the Title and the Ingredients and use that as an input to the decoder to generate the whole Instruction.

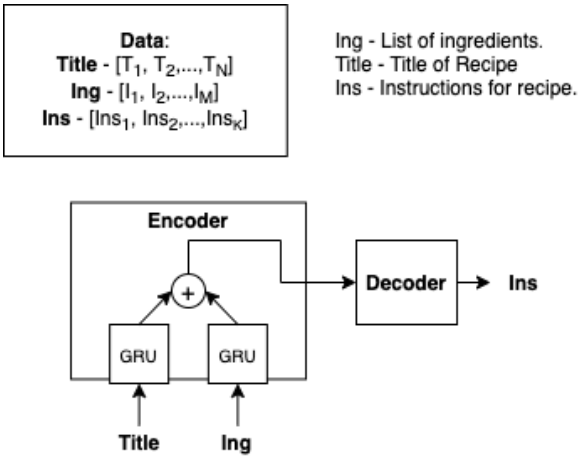

Figure 1: Seq2Seq Model

$$P(Ins|Title, Ingr; \Theta_d) = \prod_{j=1}^{T} p(y_j|Encoder(Ingr, Title; \Theta_e), y_1, ..., y_{j-1}; \Theta_d) \quad (1)$$

$$L_{s2s}(\Theta_e, \Theta_d) = -\sum_{i=1}^{T} \log P(Ins_i|Title, Ingr; \Theta_e, \Theta_d) \quad (2)$$

2.3 Seq2Seq+Attention Model

To check for comprehension we add an attention layer over the list of ingredients and the instructions generated by the decoder.

The Encoder has 2 GRU-RNN. One for the Title and the second one for the list of ingredients. We combine the hidden output for both RNN's as the combined context for our Decoder. In the attention model we also consider the outputs generated by the Ingredient RNN and add an attention layer over the outputs. We describe our model in Figure 2.

$$O_{ingr}, H_{ingr}, O_{title}, H_{title} = Encoder(Ingr, Title; \Theta_e) \quad (3)$$

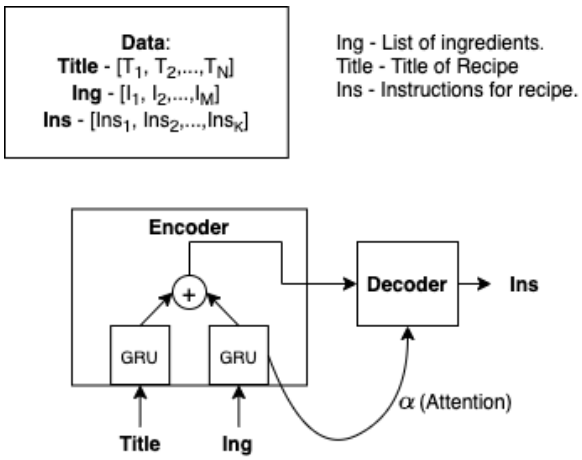

Figure 2: Seq2Seq Model with Attention

$$\alpha = Attention(O_{ingr}, h_{decoder}) \quad (4)$$

$$P(Ins|Title, Ingr, \alpha; \Theta_d) = \prod_{j=1}^{T} p(y_j|Encoder(Ingr, Title; \Theta_e), y_1, ..., y_{j-1}; \Theta_d) \quad (5)$$

$$L_{s2satt}(\Theta_e, \Theta_d) = -\sum_{i=1}^{T} \log P(Ins_i|Title, Ingr; \Theta_e, \Theta_d) \quad (6)$$

2.4 HRED+Attention Model

The attention component in the Seq2Seq model shows an increase in the models ability to be comprehensive. Hence we retain the Seq2Seq architecture while adding another Encoder and a Context-Encoder to allow the model to generate instructions in a hierarchical manner. A hierarchical generation model generates the Instructions one step at a time. The HRED with attention model is described in Figure 3.

We also incorporate a Multi-layer perceptron classifier on the decoder output. The classifier is trained to predict whether the instruction generated by the decoder at that time step is the final instruction for the recipe or not.

$$O_{ingr}, H_{ingr}, O_{title}, H_{title} = IngrEncoder(Ingr, Title; \Theta_{ei}) \quad (7)$$

$$O_{src}, H_{src} = SourceEncoder(Ins_{i-1}; \Theta_{es}) \quad (8)$$

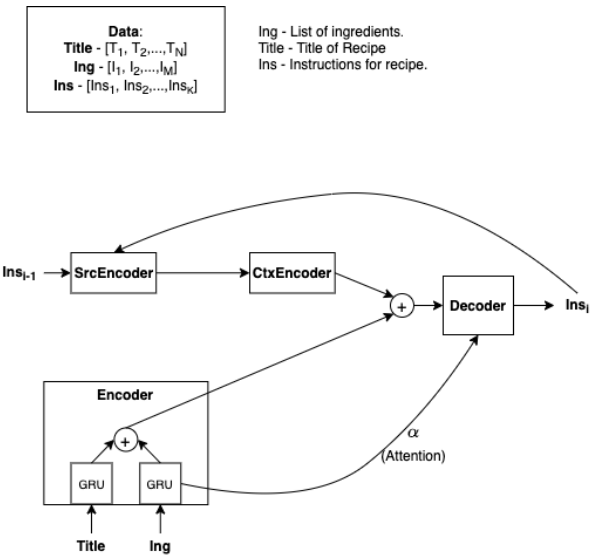

Figure 3: HRED Model

$$H_{srcCtx} = ContextEncoder(H_{src}; \Theta_{ec}) \quad (9)$$

$$\alpha = Attention(O_{ingr}, h_{decoder}) \quad (10)$$

$$P(Ins_i|Title, Ingr, \alpha; \Theta_d) = \prod_{j=1}^{T} p(y_j|ContextEncoder(Ingr, Title; \Theta_{ec}), y_1, ..., y_{j-1}; \Theta_d) \quad (11)$$

$$L_{HRED}(\Theta_{ei}, \Theta_{es}, \Theta_{ec}, \Theta_d) = -\sum_{i=1}^{T} \log P(Ins_i|Title, Ingr; \Theta_e, \Theta_d) \quad (12)$$

$$L_{isEnd}(\Theta_{ei}, \Theta_{es}, \Theta_{ec}, \Theta_d, \Theta_{class}) = -\sum_{i=1}^{M} \log\, p(e_i|Decoder(Ins) \quad (13)$$

3 Experiment

3.1 Data

We utilize the 1Million Recipe Dataset gathered by Marin Et al. for the task of Img2Seq. The goal was to generate the corresponding recipe from the image of the food item. (Marin et al., 2019)

However, considering the fact that the dataset is massive we sampled $50,000$ recipes for the training set and $5,000$ recipes for the test and validation set. We utilize the dataset parsing tool created by Salvador Et al. (Salvador et al., 2019).

Considering the small training dataset we initialize all the encoders and decoders with pretrained Glove Embeddings proposed by Pennington et al. (Pennington et al., 2014)

3.2 Evaluation

We focus or evaluation on three factors. Reconstruction, feasibility and Comprehension.

For reconstruction we focus on the BLEU score metric (Papineni et al., 2002). We also apply the method-4 smoothing function proposed by Chen et al. (Chen and Cherry, 2014).

For feasibility we create a parsed representation from the training set. We consider the adjective and verb associated with each noun phrase/token in the sentence. This is considered as the set of feasible actions that can be performed on that item. For instance, "cutting" and "onion" is considered feasible however "cutting" and "milk" is not. We define it as $(\#Correct - Incorrect)/(\#Ingredients - Generated + 1)$.

The comprehension metric checks whether the noun phrase generated by the decoder is present in the set of ingredients provided as an input to the encoder. We define it as: $(\#Ingredients - Generated)/(\#True - Ingredients)$

3.3 Results

We provide the results generated by the three models in this subsection. We only provide the results obtained by the reconstruction BLEU score since in our training it was observed that the feasibility score got an excellent score by a few sparse phrases being generated like "preheat oven" and "put the skillet on" etc. And the comprehension score proved faulty since it was close to 0 for all the generated recipes.

We provide the results obtained by the BLEU score in Figure 4 and Figure 5.

4 Conclusion

Observing the BLEU score we realise that the reconstruction is not up to the mark for the Seq2Seq models. However, the HRED model improves on the reconstruction result.

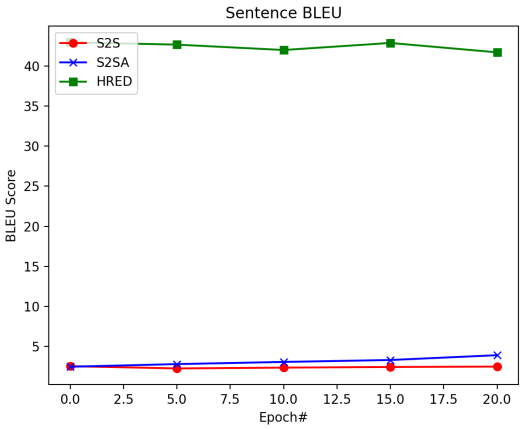

Figure 4: Sentence BLEU Score

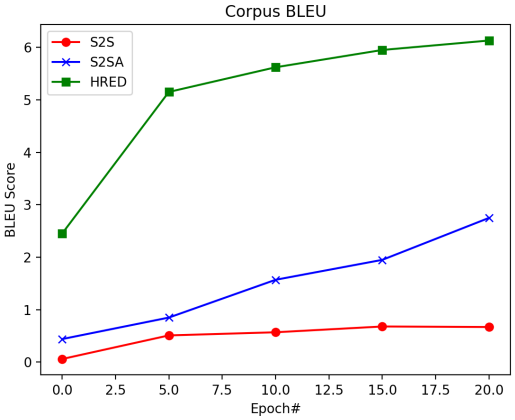

Figure 5: Corpus BLEU Score

However, the comprehension capability improves by the addition of the attention layer that focuses on the set of ingredients available for the recipe.

These baseline models provide us with two important observations. The first that generating recipe one instruction at a time is better than generating the whole recipe at once. Hence, the hierarchical model is much more suitable for our task. The second observation is the fact that the attention layer allows us to focus on the set of ingredients.

Hence, adding additional classifiers on the output of the encoder and context encoders to predict the action and ingredient being talked about in that particular recipe instruction step. For future work, apart from discrete latent variables for the ingredients and actions we can also look at variational models that can help us to sample more diverse outputs.

Also, some of the more sophisticated architec-

tures like transformers might be better served to provide us with better reconstruction.

Finally, we have to consider the fact that due to time constraints and our decision to use only $50,000$ training items our results don't correspond to the complete spectrum our models feasibility. Better and increased training might provide significant result gains as our training results show definite over-fitting on the training set.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level bleu. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 362–367.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2019. Strategies for structuring story generation. *arXiv preprint arXiv:1902.01109*.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. *arXiv preprint arXiv:1703.00955*.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. *arXiv preprint arXiv:1609.09552*.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155*.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*.

Lajanugen Logeswaran, Honglak Lee, and Samy Bengio. 2018. Content preserving text generation with attribute controls. In *Advances in Neural Information Processing Systems*, pages 5103–5113.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. 2019. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE transactions on pattern analysis and machine intelligence*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2015. Ghostwriter: Using an lstm for automatic rap lyric generation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1919–1924.

Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. 2016. Variational autoencoder for deep learning of images, labels and captions. In *Advances in neural information processing systems*, pages 2352–2360.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners.

Amaia Salvador, Michal Drozdzal, Xavier Giro-i Nieto, and Adriana Romero. 2019. Inverse cooking: Recipe generation from food images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2015. Hierarchical neural network generative models for movie dialogues. *arXiv preprint arXiv:1507.04808*, 7(8):434–441.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 553–562.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, Jun Xu, Huanhuan Cao, and Xueqi Cheng. 2018. Question headline generation for news articles. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 617–626.