Cyclegen: Cyclic consistency based product review generator from attributes

Vasu Sharma
School of Computer Science,
Carnegie Mellon University
sharma.vasu55@gmail.com

Harsh Sharma
Robotics Institute
Carnegie Mellon University
harsh.sharma@gmail.com

Ankita Bishnu
Indian Institute of Technology, Kanpur
ankita.iitk@gmail.com

Labhesh Patel
Jumio Inc.
labhesh@gmail.com

Abstract

In this paper we present an automatic review generator system which can generate personalized reviews based on the user identity, product identity and designated rating the user wishes to allot to the review. We combine this with a sentiment analysis system which performs the complimentary task of assigning ratings to reviews based purely on the textual content of the review. We introduce an additional loss term to ensure cyclic consistency of the sentiment rating of the generated review with the conditioning rating used to generate the review. The introduction of this new loss term constraints the generation space while forcing it to generate reviews adhering better to the requested rating. The use of 'soft' generation and cyclic consistency allows us to train our model in an end to end fashion. We demonstrate the working of our model on product reviews from Amazon dataset.

1 Introduction

In this age of growing e-commerce markets, reviews are taken very seriously, however, manually writing these reviews has become an extremely laborious task. This leads us to work on systems which can automatically generate realistic looking reviews which can be automatically customized to the user writing it, the product being reviewed and the desired rating the generated review should express. This makes the reviewing process much easier which can potentially increase the number of reviews posted leading to a more informed choice for potential buyers.

Natural Language Generation has always been one of the most challenging task in the field of natural language processing. Most of the present day approaches very loosely constraint the generation process often leading to ill formed or meaningless generations. Ensuring semantic and syntactic coherence across the generated sentence is also an immensely challenging task. We explore enforcing additional constraints on the generation process which we hope will restrict the generation manifold and generate more meaningful and semantically consistent sentence also adhering to the desired ratings. In this paper we attempt to perform the following tasks:

- We implement an automatic review generator using Long Short Term Memory Networks (LSTM) (Hochreiter and Schmidhuber, 1997), which has proved useful in remembering context and modelling sentence syntax. We also incorporate a soft attention mechanism which helps the model to attend better to the relevant context and generate better reviews. Such a review generator system caters to each individual users reviewing style and would convert a user provided rating into a review personalized to the users writing style and based on their rating.
- Sentiment Analysis from reviews. This includes going through the reviews and trying to gauge user sentiment and assign a score based on it. Score parameters have been found to be much easier to go through and base ones decisions upon rather than manually going through hundreds of reviews.
- In this paper we propose an additional cyclic consistency loss term which allows for joint training of the generation network with the sentiment analysis network. This improves the generator network which is now more constrained and is forced to generate reviews which adhere to the provided rating.

Proceedings of The 11th International Natural Language Generation Conference, pages 426–430,
Tilburg, The Netherlands, November 5-8, 2018.

- The use of 'soft' generation instead of a sampling based generation allows end to end gradient propagation allowing us to train our models end to end.

2 Dataset

In this paper we validate our generation framework on the Amazon dataset which contains reviews and scores for products sold on amazon.com and is part of the dataset collected by McAuley and Leskovec (2013). We used the reviews in the books category. Specifically, we have 80,256 books and 19,675 users after using the same preprocessing as used in (Dong et al., 2017). The ratings are converted into 5 integer levels from 1-5.

3 Attribute Based Review generation

In this section we explain the network we used for the attribute based review generation. The network we implement uses an architecture similar to the one proposed by Dong et al (2017). The overview of the architecture is shown in Figure 1. The architecture consists of 3 parts i.e. Attribute Encoder, Sequence Generator and a soft attention mechanism. We now describe these parts in detail.

3.1 Attribute Encoder

Let us represent the attributes by a vector a where each element of a represents a specific attribute. In our case the attribute vector consists of user ID, product ID and the rating on a scale of 1-5 each represented as one hot vectors. The model begins by using a multi layer perceptron with a single hidden layer to learn the attribute embeddings.

$$g(a_i) = W_a^i \cdot (a_i)$$

Where a_i are the one hot representations of the various attributes. This allows each of the attributes to be encoded separately. We then combine the various attribute embeddings by concatenating them and passing them through another layer of a Multi Layer Perceptron.

$$e_a = tanh(W_g \cdot [g(a_1), ..., g(a_n)] + b_a)$$

Here e_a denotes the final joint representation of the attribute embeddings and n represents the number of attributes ($n = 3$ in our case). [.] represents the concatenation operator.
The weight matrix W_g here is chosen to generate an output of size Ln where L is the number of layers in the generator network and n is the hidden state size of the LSTM units in the generator network. e_a is now used to initialize the hidden states of the multi layer LSTM based generator network.

3.2 Sequence Generator

The sequence generator network is based on a Multi Layer LSTM architecture. Unlike Dong et al, we initialize our word embeddings using a concatenation of the Glove (Pennington et al., 2014) and Cove embeddings (McCann et al., 2017). The word embeddings are fine tuned as the network trains. The attribute encodings defined in the previous section are used to initialize the hidden state of the generator network. The Ln dimensional attribute encoding is split into L parts of length n each which are used to initialize the hidden states of the L layers of the LSTM network. This basic model of the generator network without the soft attention mechanism is shown in Figure 2

3.3 Soft Attention Mechanism

Soft attention has recently been utilized to better utilize contextual information in a variety of tasks (Maas et al., 2011), (Wang and Manning, 2012). In this paper, we utilize the soft attention mechanism to make better use of the encoding information from the attributes. The architecture which implements the soft attention mechanism is shown in Figure 3. The attention is computed from the hidden vector of the LSTM over all the attribute embeddings we learned using the attribute encoder. This attention is then used to compute the attention weighted context vector c^t. This is represented by the equations:

$$r_t^i = exp(Tanh(W_h^s \cdot h_t^L + W_a^s \cdot g(a_i))$$

$$s_t^i = \frac{r_t^i}{\sum_{j=1}^{n} r_t^j}$$

$$c^t = \sum_{i=1}^{n} s_t^i \cdot g(a_i)$$

Here s_t^i is the attention weight of the i^{th} attribute and n is the number of attributes. Here W_h and W_a are parameter matrices. We use this attention weighted context vector to predict the next word

generated by the sequence generator as:

$$h_t^{att} = tanh(W_1 \cdot c_t + W_2 \cdot h_t^L)$$
$$o_t = (W_p \cdot h_t^{att})$$

Here W_1, W_2 and W_p are parameter matrices. The generation thus involves a sequence of discrete decision making which samples a token from a multinomial distribution parameterized using softmax function at each time step t:

$$\hat{x}_t \sim softmax(o_t/\tau)$$

where o_t is the logit vector as the inputs to the softmax function. The temperature τ is set to $\tau \to 0$ as training proceeds, yielding increasingly peaked distributions that finally emulate discrete case. The generation process ends when the EOS token is generated or when 3 complete sentences are generated, whichever happens first.

4 Training

The review network is initially pre-trained independently of the sentiment analysis network by maximizing the log likelihood of the generated sequence. After running a few epochs of training the generator alone, we enforce an additional cyclic consistency term in the loss function. The idea is the sentiment analysis score of the generated review should be consistent with the original rating provided as an attribute. Similar consistency terms can be applied to the other attributes as well, but here we explore only the consistency of the rating score term. A cross entropy loss between the predicted sentiment rating class and the ground truth rating class is used as the additional loss function to enforce cyclic consistency. Since sampling words from the generator will make the model non-differentiable preventing end to end training, hence we keep things in the probabilistic domain by resorting to a continuous approximation by using the probability vector instead of the sampled one hot vector. The probability vector is used as the output at the current step and the input to the next step along the sequence of decision making. This leads to a 'soft' predicted sequence $\widetilde{G}(a)$, which we use to compute the cyclic rating consistency loss term and this being fully probabilistic is differentiable allowing end to end training of the network. The cyclic consistency loss term can be denoted as:

$$L_{cyc} = \mathbb{E}_{(a,r)\in D}\, q_D(\widetilde{G}(a), r)$$

where q_D is the loss from the sentiment rating class predictor and r is the ground truth rating. Hence the joint loss function becomes:

$$L_{tot} = -L_{likelihood} + \lambda L_{cyc}$$

Adam optimizer (Kingma and Ba, 2014) with default parameters is used to train the model. NLTK tokenizer (Bird et al., 2009) is used to tokenize the sentences and all words which appear less than 10 times in the corpus are replaced by the $< UNK >$ token. All LSTM layers use 512 dimensional hidden units and 3 layers are used in the generator LSTM. The test time generations are generated using greedy search algorithms.

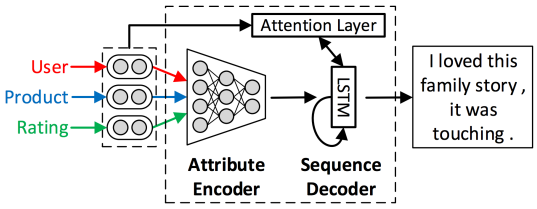

Figure 1: The model first learns attribute embeddings and then uses an LSTM network to generate the reviews one word at a time. A soft attention mechanism is used to learn alignments between attribute embeddings and the generated words.

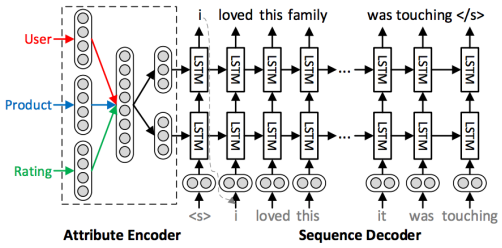

Figure 2: The basic setup of the generator network without attention

5 Sentiment Analysis Rating Predictor

For sentiment analysis we use a bidirectional RNN with Gated Recurrent Units (GRU) (Chung et al., 2014) pipeline which takes as input the generated review and generates a rating score at the end. The words are first embedded to vectors using an embedding layer which is initialized using

User	Product	Rating	Generated Review
A	X	1	the story was really boring. i was expecting much more. the ending was abrupt.
A	X	5	i loved the characters. the movie was thoroughly enjoyable. the plot was well written.
B	X	1	this book is not as good as the previous one. i was looking forward to reading this. i will not be reading the next one.
B	X	5	the books from this author just keep getting better. will highly recommend this book to everyone. looking forward to more books from the author.
A	Y	1	the plot of this book is really confusing. the book is not well written. i was unable to read the whole book.
A	Y	5	i really enjoyed reading this book. the characters are amazing.

Table 1: Some examples from the review generator network for various users, products and rating scores

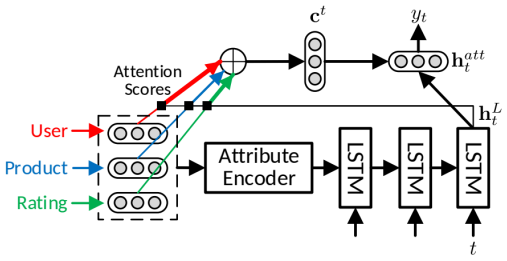

Figure 3: The soft attention is computed by using the present hidden state of the generator LSTM and the attribute vector. Attention weighted attribute vector embeddings are used as input to the generator along with the previous generated word to generate the review.

Method	BLEU-4 (%)	BLEU-1 (%)
Rand	0.86	20.36
MELM	1.28	21.59
NNpr	1.53	22.44
NNur	3.61	26.37
Att2Seq	4.51	30.24
Att2Seq+A	5.03	30.48
Cyclegen(Ours)	**5.46**	**30.63**

Table 2: Evaluation of our generated sentence quality using BLEU score and comparison with baseline systems (details in Appendix A)
Baseline results as in (Dong et al., 2017)

Method	Att2seq	Att2seq+A	CycleGen
Accuracy(%)	82.3	85.6	87.5

Table 3: Accuracy of polarity (positive/negative) of the generated sentences by manual human comparison against input polarities (1-3 is considered negative and 4-5 is considered positive)

concatenations of Glove embeddings (Pennington et al., 2014) and CoVe embeddings (McCann et al., 2017). We noticed a substantial performance improvement by using the CoVe embeddings in addition to the Glove embeddings compared to the traditionally used Glove embeddings or word2vec embeddings (Mikolov et al., 2013). We also allow the embeddings to be fine tuned with training.
The final hidden layer generated by the GRU is then passed onto a Multi Layer Perceptron which finally predicts the sentiment rating class.

6 Results

After a sufficient amount of training the network learns to generate some realistic looking reviews. The additional loss term seems to force the review to not be repetitive and not to use generic words besides ensuring that the generated review adheres to the expected rating. For evaluation of the generated sentence quality, we use BLEU score which measures the precision of n-gram matching by comparing the generated results with references, and penalizes length using a brevity penalty term. Here we use BLEU-1 (unigram) and BLEU-4 (upto 4 grams) to evaluate our models. The results for the same and comparison with some other works on the same task are shown in Table 2. We also perform some human evaluation of the polarity of the generated reviews against input polarity. The results for the same are shown in Table 3.

We also notice that the baseline sentiment analysis rating system which was trained directly on the Amazon reviews dataset attained an accuracy of 70.1% which improves to 72.4% when fine-tuned using this end to end framework. Some of the reviews generated by the system and their corresponding ratings are demonstrated in the Table 1

References

S. Bird, L. Edward, and K. Ewan. 2009. Natural language processing with python. *O'Reilly Media Inc*.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.

Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to generate product reviews from attributes. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 1.

Michael U. Gutmann and Aapo Hyvärinen. 2012. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *J. Mach. Learn. Res.*, 13(1):307–361.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

A. Maas, R. Daly, Peter Pham, D. Huang, A. Ng, and C. Potts. 2011. Learning word vectors for sentiment analysis. *In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*.

J. McAuley and J. Leskovec. 2013. Hidden factors and hidden topics: Understanding rating dimensions with review text. *In Conference on Recommender Systems, RecSys*, pages 165–172.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. *Advances in Neural Information Processing Systems 30 (NIPS 2017)*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation.

S. Wang and C. Manning. 2012. Baselines and bigrams: Simple, good sentiment and text classification. *In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*.

A Details of Baseline Systems

We describe the comparison methods as follows. Note that the comparison baselines are same as used in (Dong et al., 2017):

- **Rand**: The predicted results are randomly sampled from all the reviews in the TRAIN set. This baseline method suggests the expected lower bound for this task.
- **MELM**: Maximum Entropy Language Model uses n-gram (up to trigram) features, and the feature template attribute n-gram (up to bigram). The feature hashing technique is employed to reduce memory usage in each feature group. Noise contrastive estimation (Gutmann and Hyvärinen, 2012) is used to accelerate the training by dropping the normalization term, with 20 contrastive samples in training.
- **NN-pr**: This Nearest Neighbor based method retrieves the reviews that have the same product ID and rating as the input attributes in the TRAIN set. Then we randomly choose a review from them, and use it as the prediction.
- **NN-ur**: The same method as NN-pr but uses both user ID and rating to retrieve candidate reviews
- **Att2seq**: The basic LSTM encoder decoder model without any attention mechanism.
- **Att2seq+A**: The present state of the art model on this task as explained in (Dong et al., 2017)