Dialogue Generation Versus Sequence to Sequence Learning

Saeed Najafi
Department of Computing Science
University of Alberta
Edmonton, Canada
snajafi@ualberta.ca

Abstract

In this work, we investigate sequence to sequence models for dialogue generation where given an utterance, the model should generate a relevant response. We suggest that there are three underlying obstacles which are inherent to these models. First, the learning objective in the sequence to sequence model does not consider the relevance of the response to the given utterance and only forces the model to learn the response of the training example. Second, the OpenSubtitles dataset is not suitable for building a conversation system. Finally, the attention mechanism, introduced for machine translation, does not provide any improvements over the vanilla sequence to sequence model in the dialogue generation task.

1 Introduction

One of the Artificial Intelligence objectives, especially in Natural Language Processing, is to have an intelligent computer program that can communicate well with humans. Companies like Google, Apple and Microsoft make a great investment to build their intelligent personal assistants in their products (known as Google Now, Siri and Cortana). Similarly, but in a specific domain, Quinn developed an automatic nursing agent (ANA) that operates as a companion to the elderly. ANA can assist the elderly in information retrieval and can remind them of relevant events (related to health). One of the difficult challenges for ANA is producing machine-generated responses that are indistinguishable from human spoken language.

Recently, Sutskever et al. introduce an end-to-end model for machine translation, namely, Sequence to Sequence (Seq2Seq) model. Sordoni et al. and Serban et al. explore how this Seq2Seq model can be used for open domain dialogue generation. In the same direction, Li et al. develop a persona-based conversation system using the Seq2Seq model. Others try to modify the model to generate multiple responses to a dialogue utterance and then, propose re-ranking techniques to select a more relevant response from this generated list (Li et al., 2016b), (Li et al., 2015) and (Li et al., 2017).

This work investigates the Seq2Seq model for dialogue generation and highlights three underlying obstacles which make the model impractical to get integrated in a conversation system such as ANA.

- The learning objective function in the Seq2Seq model is not suitable for learning and generating relevant responses, especially for datasets in which there is only one response to an utterance.
- The OpenSubtitles[1] corpus, which is widely used in the literature for training and evaluating dialogue generation models, is too open domain, and therefore, inappropriate for this task.
- Attention mechanism (Bahdanau et al., 2014), which empowers the Seq2Seq model for machine translation, does not offer significant improvement over the vanilla Seq2Seq model for generating relevant responses.

In the following sections, the background of the Seq2Seq model is introduced in section 2 and the model itself is explained in section 3. Section 4 discusses the conducted experiments and finally, some future directions are mentioned in section 5.

[1] http://opus.lingfil.uu.se/OpenSubtitles.php

2 Background

2.1 Semantic Vectors

Semantic vectors are created by mapping words into limited-in-dimension vectors. Word2vec, introduced by Mikolov et al., uses a neural network to generate these vectors. In the skip-gram architecture of word2vec, a neural network is trained on a large text corpus to predict the surrounding words of each word. Two words, which co-occur frequently, tend to have similar semantic vectors in the vector space.

2.2 Recurrent Neural Networks (RNN)

The main difference of a recurrent neural unit with a simple neuron is that its previous output can be used to calculate the next output in the following time step. In a simple neuron, the output is only calculated based on the input. On the other side, in a recurrent neural unit, not only does the output depend on the input, but also it relies on the previous output. The following formulas represent a simple neuron and a recurrent neural unit with $tanh$ activation functions where I_t is the input of the neuron at the time step t, U and UU are weight matrices, and b is the bias vector.

Simple neuron:

$$O_t = tanh(I_t \times U + b)$$

Recurrent neuron:

$$O_t = tanh(I_t \times U + b + O_{t-1} \times UU)$$

A recurrent unit can be used to model a sequence according to which every output of the neuron represents one state in the sequence. The previous output of the neuron has information about the past steps of the sequence and in the case of a sentence, this information is about previously seen words. In a sentence, the semantic vectors of words are given as inputs to a recurrent unit. For example, in the sentence S which contains N words $W_1, ..., W_t, ..., W_N$, the semantic vector of W_t, which is $SV(W_t)$, is fed into the unit at the time step t. In the last time step, O_N semantically encodes the whole sentence.

Because of the vanishing gradient problem of the back propagation method in training RNNs (Pascanu et al., 2012), the default recurrent unit can only keep information of the past few steps. To make a recurrent unit remember longer steps, Chung et al. introduce Gated Recurrent Units (GRU) and compare it with Long Short-Term Memory (LSTM) units introduced by Hochreiter and Schmidhuber. The mathematical notation of an LSTM unit is as follows:

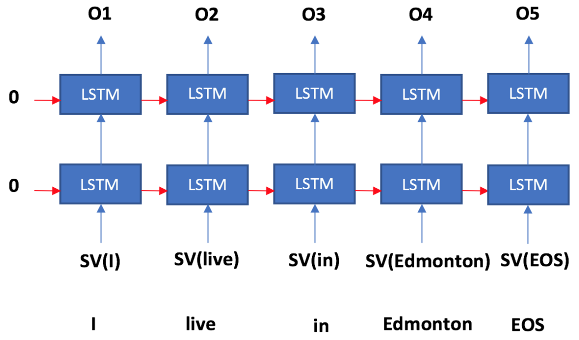

Figure 1: Unfolded two layer recurrent neural network with LSTM units. SV(W) is the semantic vector of W.

$$IG_t = sigmoid(I_t \times U_1 + O_{t-1} \times UU_1 + b_1) \quad (1)$$

$$FG_t = sigmoid(I_t \times U_2 + O_{t-1} \times UU_2 + b_2) \quad (2)$$

$$EG_t = sigmoid(I_t \times U_3 + O_{t-1} \times UU_3 + b_3) \quad (3)$$

$$NM_t = tanh(I_t \times U_4 + O_{t-1} \times UU_4 + b_4) \quad (4)$$

$$M_t = FG_t \cdot M_{t-1} + IG_t \cdot NM_t \quad (5)$$

$$O_t = EG_t \cdot tanh(M_t) \quad (6)$$

IG: Input Gate, FG: Forget Gate, EG: Exposure Gate, NM: New Memory, M: Memory, O: Output, and I: Input

An LSTM unit can be stacked on top of another LSTM unit such that the output of the unit in the first layer is fed as an input to the next layer. It has been shown that a stacked RNN performs better than a single layer RNN in sequence to sequence models for machine translation (Sutskever et al., 2014). A stacked RNN with LSTM units is illustrated in Figure 1.

3 Sequence to Sequence Models

The Seq2Seq model is first introduced by Sutskever et al. for machine translation. In this model, for an input sequence, another sequence is generated. For example, in machine translation, the former sequence can be a sentence in English and the target sequence can be its translation in French. The translation begins by giving an English sentence to an encoder which embeds the sentence into a limited-in-dimension vector. The encoder is usually a recurrent neural network in which its output in the final step is considered as the embedding for the sentence. As illustrated in Figure 1, the output $O5$ is an embedding or context vector for the input sentence “I live in Edmonton”.

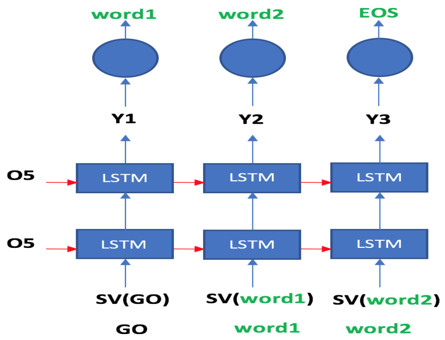

Figure 2: Decoder recurrent neural network with LSTM units. SV(W) is the semantic vector of W. O5 is the embedding or context vector from the encoder RNN.

For generating the target sentence, which is the French translation of the input sentence in this example, another recurrent neural unit is used which is initialized with the resulting embedding vector from the encoder RNN. To use an RNN for generating a sentence, a probability distribution function is required which predicts the probability of W_t being generated based on the output of the decoder RNN in the time step t. Therefore, an output layer is necessary over the decoder RNN to calculate these probabilities for words. This output layer can be formulated in the following equation where $softmax$ function is used to have a probability distribution over the vocabulary set ($softmax(Y) = \frac{e^{y_j}}{\sum_{i=0}^{|V|} e^{y_j}}$).

$$Y_t = softmax(O_t \times U + b) \quad (7)$$

$$word_t = \arg\max(Y_t) \quad (8)$$

U is a $d \times |V|$ weight matrix where $|V|$ is the vocabulary size and d is the size of O_t. b is a $|V| \times 1$ bias vector. $word_t$ is the generated word in the time step t.

In the decoder RNN, the generated word from the previous step is fed into the LSTM unit to generate the next word (displayed in Figure 2). The initial word is the special symbol 'GO' which triggers the decoder to generate the target sequence. It should be pointed out that the length of the target sequence can be different from the input sequence. Both encoder and decoder RNNs are trained together with respect to the following objective function where S ($S = W_1, W_2, ..., W_{|S|}$) and T ($T = T_1, T_2, ..., T_{|T|}$) are the input and target sequences (sentences) respectively.

Objective function $P(T|S)$:

$$P(T|S) = \prod_{k=1}^{|T|} p(T_k|W_1, ..., W_{|S|}, T_1, ..., T_{k-1})$$

$$P(T|S) = \prod_{k=1}^{|T|} Y_k$$

The Seq2Seq model has been used for many NLP tasks such as machine translation (Sutskever et al., 2014), grapheme-to-phoneme conversion (Yao and Zweig, 2015), text summarization (Nallapati et al., 2016) and dialogue generation (Sordoni et al., 2015). For dialogue generation S and T can be dialogue utterances between two entities.

Bahdanau et al. introduce attention mechanism to the Seq2Seq model according to which instead of using a fixed embedding vector, Y_t is calculated based on a separate context vector c_t which is computed as the weighted average of all the outputs of the encoder RNN ($c_t = \sum_{j=1}^{|S|} \alpha_{t,j} \times O_j$). $\alpha_{t,j}$ is given by an alignment model which scores how well the input words around position j and the generated words at position t match. The Seq2Seq model with attention mechanism has been shown to perform better for machine translation because it also learns an alignment model for handling various orderings of words in the input and target sentences.

4 Experiments

In order to conduct experiments, 12 million dialogue utterances are selected from the OpenSubtitles corpus (Tiedemann, 2009). In the preprocessing step, dialogues of length 2 are chosen (one utterance and one response, each with minimum 6 words). The development set is 10% of this dataset and 50000 utterances-responses are held out from this dataset as the test set. In addition to the quantitative evaluation, 50 dialogues about 10 topics are designed for the qualitative analysis (5 dialogues for each topic). The topics include "family", "food", "job & work", "relationships", "crime", "school", "nature", "health", "greetings", and "feelings".

4.1 Experiments Setup

For evaluation, the sentence-level BLEU is used to measure how models can exactly generate the same responses of the test set. The BLEU metric correlates well with the human judgment and is widely used for automatic analysis of machine translation methods (Papineni et al., 2002). It is noteworthy that the BLEU metric does not check how relevant the generated response is to the given utterance. Therefore, in the qualitative evaluation, two other metrics are utilized: the Relevance Rate (RR) and the Diversity Rate (DR). For calculating the Relevance Rate, first, the generated response

Model	Avg. BLEU	RR	DR
Attention Seq2Seq	0.134	10%	60%
Vanilla Seq2Seq	0.131	11%	62%
Shuffled, Seq2Seq	0.123	4%	48%

Table 1: Average sentence-level BLEU ([0,1]) on the test set and the relevance rate (RR) and the diversity rate (DR) of the qualitative evaluation.

is manually given a score of 0, 0.5 or 1 (1 corresponds to the completely relevant response). Afterward, the scores for all responses are averaged to compute the Relevance Rate. The Diversity Rate is simply the percentage of the generated responses that are unique. Qualitative analysis with RR and DR metrics are only conducted on the 50 designed dialogues.

4.2 Evaluation & Conclusion

In the first set of experiments, the vanilla Seq2Seq is compared to the Seq2Seq model with the attention mechanism. The vanilla model was trained 4 epochs on the dataset and its perplexity over the development set could reach 30 ($perplexity = e^{loss}$). The model with the attention mechanism was trained only 2 epochs, but it also reached the perplexity of 30. The training time for both models was about 6 hours on a Tesla K80 GPU.

As presented in Table 1, The attention mechanism does not improve the performance of the Seq2Seq model for dialogue generation. Although the Seq2Seq model with attention mechanism converges to the perplexity of 30 only in two epochs, it requires more training time compared to the Vanilla Seq2Seq model (roughly 2 times more). The vanilla Seq2Seq reaches the same perplexity in 4 epochs with the same training time as of the attention mechanism with 2 epochs. This extrinsic evaluation within the dialogue generation task supports the claim that the attention mechanism does not empower the Seq2Seq model for dialogue generation, although it is helpful for a translation model.

For the next experiment, responses of the training dataset are randomly shuffled and a Vanilla Seq2Seq model is trained using this random dataset (the responses does not make any sense in this new dataset). Table 1 demonstrates that the Seq2Seq model preforms better on the true dataset than on a randomly shuffled dataset (with more relevant and diverse responses), however, the performance difference suggests that the true dataset is too open domain, therefore, not appropriate for training a dialogue generation model using sequence to sequence learning.

Model	Utterance | Response
Vanilla	we are going to kill him tonight in his car | i ’ m not going to do anything
Vanilla	How should I propose her? | i ’ m not going to do it
Vanilla	Don’t forget that you have a job interview tommorow. | you ’ r e a good man
Vanilla	I hate people who even think about suicide. | i don ’ t know what to do
Vanilla	I really miss you. | i UNknown l be right back
Attention	They are so disappointed! | what ’ s wrong ?
Attention	Would you like some coffee? | i ’ m not gonna get you a drink
Attention	How should I propose her? | i ’ m not going to tell you
Attention	Nothing is worse than a stupid fight with your brother! | i ’ m not a fool
Attention	My boss is a very serious guy. | he ’ s a very good man
Shuffled	Would you pass the bottle of water to me? | that ’ s the way i want
Shuffled	I will help you to find a job. | i UNknown l be right back
Shuffled	My boss is a very serious guy. | i ’ m not sure he ’ s a good man

Table 2: Relevant responses of the vanilla Seq2Seq, Seq2Seq with attention mechanism and the vanilla Seq2Seq trained on the shuffled dataset

The main reason for 4% relevance rate for the Seq2Seq model trained on the randomly shuffled dataset is that the model generates common responses such as “I do not know” or “I am not sure”. Therefore, in some cases, these common responses can be relevant for a given utterance. As an example, for the utterance “What should I do?”, the common response “I do not know” is relevant. It should be pointed out that the vanilla Seq2Seq trained on the true dataset learns these common responses and the same model on the shuffled dataset also generates these common responses. Relevant responses of the models for the designed dialogues are shown in table 2. None of these responses has been seen in the training dataset and almost all are grammatically correct.

5 Future Work

Li et al. suggest that a beam search decoding can be used in the decoder RNN in the generation step instead of the greedy selection approach where the word with the highest probability is selected in each time step. They also suggest that penalizing and re-ranking responses based on mutual information can avoid the common responses (Li et al., 2015). However, beam search decoding and re-ranking techniques do not ensure that the the final response will be relevant to the given utterance. Also, beam search decoding and the reranking techniques are all happen in the testing phase when a response should be generated for a given utterance. Considering the objective function used in Seq2Seq models (discussed in section 3), during the training step, the models are forced to produce exactly one response which is the response in the training example.

This work proposes this idea that having a dialogue dataset where there are multiple possible responses for one utterance and modifying the objective function in the Seq2Seq model to consider all relevant responses at once, can improve the performance of the model in generating relevant responses even with the greedy response generation approach in the testing phase. Further investigation of this idea remains as the main future direction of this work.

References

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

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* pages 1735–1780.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *CoRR* .

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A persona-based neural conversation model. *CoRR* .

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016b. Deep reinforcement learning for dialogue generation. *CoRR* .

Jiwei Li, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. *CoRR* .

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Ramesh Nallapati, Bing Xiang, and Bowen Zhou. 2016. Sequence-to-sequence rnns for text summarization. *CoRR* .

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 on Association for Computational Linguistics*. ACL '02, pages 311–318.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2012. Understanding the exploding gradient problem. *CoRR* .

Kevin Francis Quinn. 2014. Ana: Automated nursing agent.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2015. Hierarchical neural network generative models for movie dialogues. *CoRR* .

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. *CoRR* .

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112.

Jörg Tiedemann. 2009. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In *Recent Advances in Natural Language Processing*, John Benjamins, Amsterdam/Philadelphia, pages 237–248.

Kaisheng Yao and Geoffrey Zweig. 2015. Sequence-to-sequence neural net models for grapheme-to-phoneme conversion. *CoRR* .