Neural Conversational Model with Mutual Information Ranking

Harrison Ho
harrisonho@stanford.edu

Chenye Zhu
chenye@stanford.edu

Abstract

We build a neural conversation system using a deep LST Seq2Seq model with an attention mechanism applied on the decoder. We further improve our system by introducing beam search and re-ranking with a Mutual Information objective function method to search for relevant and coherent responses. We find that both models achieve reasonable results after being trained on a domain-specific dataset and are able to pick up contextual information specific to the dataset. The second model, in particular, has promise with addressing the "I don't know" problem and de-prioritizing over-generic responses.

1 Introduction

Chatbots, programs that simulate human beings in conversations, have been gaining traction in the tech community in recent years. Ever since the first Verbot, Julia was created by Michael Maudlin in 1994, researchers have made various attempts to design and implement such programs. Researchers have sought out the proper configurations for a chatbot to exhibit intelligent behaviors equivalent to or indistinguishable from those of a human, and thus to pass the well known Turing test.

In general however, holding meaningful conversations with humans is hard. Traditional statistical models often struggle to understand humans' meanings and intentions (semantic information) in addition to the syntactic structures of their remarks (pattern matching). With recent developments in deep neural networks and recurrent neural networks, researchers have new advanced tools to work in this field and have developed new strategies to tackle automatic dialogue generation.

In this paper, we apply a deep LSTM Seq2Seq model with an attention-based decoder in order to tackle this task of dialogue generation. In addition, we improve on this model by applying beam search over mutual information between statement and responses to rank optimal responses. We find that using mutual information improves model outputs by de-prioritizing generic responses.

2 Previous Work

Researchers have developed many neural dialogue systems in the past. The vast majority of such neural systems use Seq2Seq as a backbone, which allows for mappings from entire sequences of words or characters to other sequences. In their paper "A Neural Conversational Model", Vinyals et. al [8] demonstrate their neural dialogue system on several datasets. In particular, they show that their model successfully incorporates contextual information in conversation, such as discussing operating systems when trained on an IT help desk training set.

Some previous work has been done with re-ranking Seq2Seq model outputs using additional syntactic and semantic features. Li et al, in their paper "A Diversity-Promoting Objective Function for Neural Conversation Models" [3], describe how usual Seq2Seq networks approximate the probability of a target sentence given a source sentence. They augment this by linearly combining the

Seq2Seq model output with an *anti*-language model, and punish over-generic responses generated by their conversation model.

In their paper "Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models", Serban et. al [7] show how using dialogue-by-dialogue level encoding and decoding, in addition to word-by-word level encoding and decoding, can lead to a model competitive with the state of the art. Serban et. al make use of a hierarchical recurrent encoder-decoder to predict individual dialogues utilizing a context RNN, which has previously been used for predicting search queries given a history of queries.

Some researchers have made advancements using reinforcement learning to improve their models. Li et. al, in their paper "Dialogue Learning with Human-in-the-Loop" [4], describe how reinforcement learning can improve the process of learning and yield better results compared with training on static datasets.

Work has also done to apply the Generative Adversarial Network framework, which has previously had success with image processing, to neural dialogue generation. Li et. al [5] demonstrate some success of their adversarial model; however, they concede that more work is required with GANs to yield significant performance boosts in the realm of natural language processing.

3 Dataset

We use the Ubuntu Dialogue Corpus [6], a collection of two way multi-turn dialogues. It contains almost 1 million dialogues, with over 7 million utterances and 100 million words. For efficiency, we only consider the 40,000 most commonly used words within the corpus, and replace other words with a special *_UNK* token.

For simplicity, our model does not make use of conversation-level features. We instead split each of the dialogues into pairs of statements and responses (e.g. a statement by person A, followed by a response by person B). As shown in table 1, the median statement length is 16, and the median corresponding response length is 14. From the descriptive statistics presented below in table 2, we see no obvious association between statement length and response length.

This dataset is notable in that it includes domain-specific knowledge relevant to Ubuntu and IT, which we find influences the quality of the dialogues generated by our model. For example, the dataset includes many question-answer pairs due to its IT help desk characteristics. As a result, our model is more capable of handling question inputs as opposed to pure statement inputs.

	25^{th} percentile	50^{th} percentile	75^{th} percentile
Statement Length	9	16	28
Response Length	7	14	24

Table 1: Statement and response length distribution within the Ubuntu Dialogue Corpus

		Statement Length			
		$1-9$	$10-16$	$17-28$	≥ 29
Response Length	$1-7$	148191	110588	97043	89374
	$8-14$	142557	118785	117023	120353
	$15-24$	111244	97935	100397	102873
	≥ 25	103624	94933	102708	114497

Table 2: Correlation between statement and response length

4 Model

We develop two models in this paper. The first is a baseline deep LSTM Seq2Seq model with an attention mechanism applied to the decoder. The second is a more advanced model which builds on the first by examining the mutual information contribution between statements and predicted responses.

4.1 Baseline Model

Our baseline model is a 3-layered deep LSTM Seq2Seq model using 500-dimensional randomly initialized word vectors and a 500-dimensional hidden state. The input statement is padded with a special padding character *_PAD* to the maximum length before feeding into the Seq2Seq model. The output is generated word by word, until a special character *_EOU*, end of utterance is generated. The Seq2Seq model is well-suited for the dialogue generation task, as it enables end-to-end training and evaluation between input and output sentences.

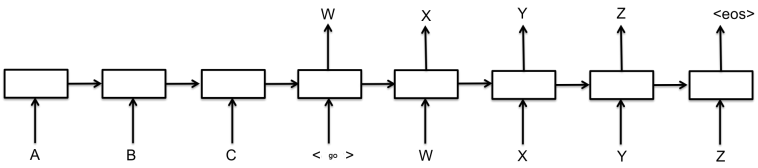

Figure 1: An example Seq2Seq architecture [1]

We apply a static dropout mask with a 50% dropout rate on each RNN cell during training to prevent overfitting; the same dropout rate is used for each of the LSTM cells regardless of the timesteps. This dropout is especially critical for dialogue generation. Neural dialogue systems can be susceptible to the so-called *"I don't know"* problem [7], where such systems repeatedly utter generic phrases such as "I don't know", which are simply statistically likely to appear and not necessarily useful. Dropout improves the situation by coercing RNN units to understand more complicated syntactic and semantic features.

Next, we apply an attention mechanism on the decoder. This allows the decoder to predict outputs not only by using the hidden states of the decoder RNN of the Seq2Seq model, but also by using an additional context state to keep track of information possibly more relevant at certain timesteps. The attention decoder is based off of improvements on the Seq2Seq decoder as explored by Bahdanau et. al [2].

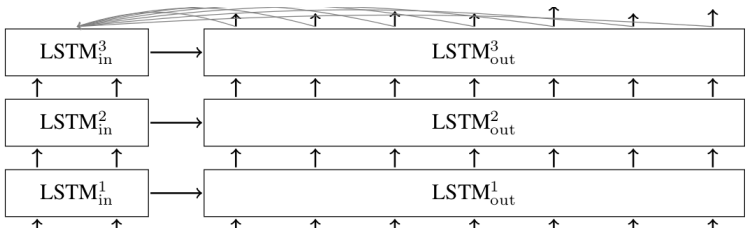

Figure 2: An example decoder architecture using attention [1]

Finally, to train the model more efficiently and balance the length of real inputs versus padding characters, we split the dataset examples into four buckets. For example, a training example pair with a statement length of 9 and a response length of 11 would be placed in the second bucket, and both the statement and response will be padded with a special *_PAD* token to reach 15 words each. The bucket sizes that we used for this model are listed in table 3.

To evaluate our model, we compute the perplexity of the model over our validation sets, and examine the perplexity for each bucket. We also perform qualitative analyses for a *randomly* selected test set to examine both grammatical soundness and logical relevance of our model outputs.

4.2 Re-ranking using Mutual Information Contribution Score

Dialogue generation systems often suffer from the excessive output of generic default safety responses, such as "I don't know". This is often because such responses are the most statistically likely to appear in a training set. Unique answers are usually specific to individual questions, making it challenging for models to learn the intricate patterns and logic correlations between input statements and output responses. Instead, the models default back to the most common responses in

Bucket Number	Criteria	Number of Training Examples
1	statement length no greater than 10 response length no greater than 10	246,774
2	statement length no greater than 15 response length no greater than 15	273,213
3	statement length no greater than 20 response length no greater than 20	262,863
4	statement length no greater than 30 response length no greater than 30	399,524

Table 3: Descriptive statistics for each bucket.

the dataset "I don't know" (or "what is the problem", as we have observed for the Ubuntu dataset). We approach this problem in our advanced model and add on a beam search mechanism to search for responses with the highest mutual information contribution scores.

We observe that the base Seq2Seq model, in essence, computes the probability of observing a response r given an input statement s_i with probability $P(r|s_i)$. It then tries to find the response $\hat{r}_i$ that maximizes the conditional probability $\hat{r}_i = \arg\max_{r \in R} P(r|s_i)$.

However as we have seen, for a substantial number of statements s_i, $\hat{r}_i$ is rather generic, such as "I don't know" or "what is the problem". In order to capture the high level logic correlation between user inputs and proper output responses, we reformulate our objective function as using the mutual information statistic between the statement and response pair (s_i, r_i), accounting for both the probability of observing the response given the input statement $P(r_i|s_i)$, and the probability of the input statement given the output response $P(s_i|r_i)$.

Notice that a statement-response pair (s_i, r_i) contributes

$$\begin{aligned} MI(s_i, r_i) &= P(s_i, r_i) \ln(\frac{P(s_i, r_i)}{P(s_i)P(r_i)}) \\ &= P(s_i)P(r_i|s_i) \ln(\frac{P(r_i|s_i)P(s_i)}{P(s_i)P(r_i)}) \\ &\propto P(r_i|s_i)(\ln P(r_i|s_i) - \ln P(r_i)) \end{aligned}$$

Hence, our new desired choice of response $\hat{r}_i$ is

$$\begin{aligned} \hat{r}_i &= \arg\max_{r \in \mathcal{R}} MI(s_i, r) \\ &= \arg\max_{r \in \mathcal{R}} P(r|s_i)(\ln P(r|s_i) - \ln P(r)) \end{aligned}$$

the response (out of all possible response sequences) that maximizes the mutual information contribution score. Here, $P(r|s_i)$ can be computed from our previous Seq2Seq model. We can compute $P(r)$ using a probabilistic language model that assigns cost measurements to sentences. For our advanced model we utilized a reproduction of a pre-existing language model developed by Zaremba, et. al [9], and trained this language model on the same Ubuntu Dialogue Corpus of which the Seq2Seq model is trained. Since generic responses are fairly common within the dataset, we would expect the language model to assign a relatively low score to these responses, and the mutual information objective function would punish such outcomes.

One practical concern that we have to resolve is that searching over the enormous response space $\mathcal{R}$ is virtually impossible. Thus, we first perform beam search on the Seq2Seq model for the top results that maximize the first part $P(r|s_i) \ln P(r|s_i)$. In the second step, utilizing the results from the language model, we *re-rank* them to achieve the top responses under the mutual information objective function. Due to limitations of time and resources, we set our beam size to 50, as compared to 800 used in Li's paper [3].

5 Experiments

5.1 Perplexity Analyses of Baseline Model

We evaluated our first model using perplexity, a measure of prediction error. A higher perplexity indicates that a particular sentence is more difficult to predict and indicates a greater degree of error. We display the achieved perplexity of the model for each of the buckets after 50,000 training iterations in table 4, and the training and dev perplexities over time in figure 3.

Number	Maximum Word Length	Perplexity
1	10	11.46
2	15	20.85
3	20	27.10
4	30	30.84

Table 4: Perplexity after 50,000 training iterations

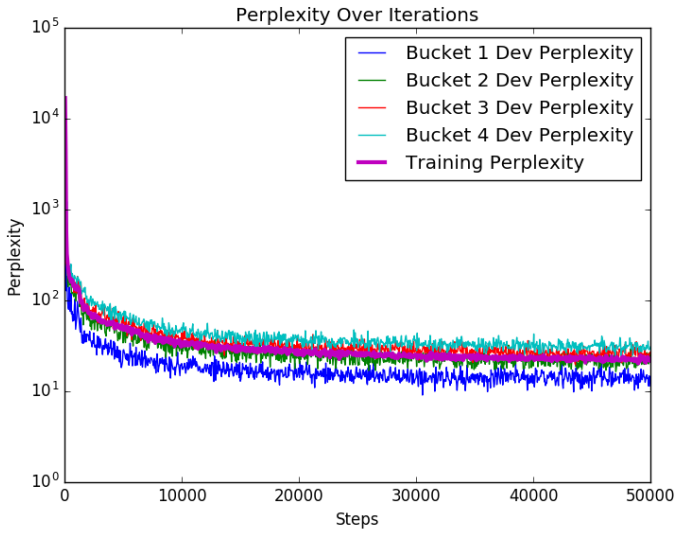

Figure 3: Training and dev perplexity over training iterations

As we can see, the validation perplexity for bucket 1 is the smallest, while the validation perplexity for bucket 4 is the greatest. This makes intuitive sense; for longer statement and response pairs, it is more difficult for the decoder to apply attention to the appropriate section of a statement to determine the appropriate output. In addition, for longer sequences, relevant information in the hidden state may not be persisted well through the LSTM.

In addition, the training perplexity over all time steps lies at approximately the average of the perplexities for each of the individual buckets. This suggests that the model is not overfitting to the training set after convergence.

5.2 Qualitative Analysis of Both Models

We qualitatively analyze our model outputs over a set of selected validation inputs to determine the qualities of our models. Two characteristic examples are presented here for detailed analysis.

In our first example, the model demonstrates that it has learned contextual information from the Ubuntu Dialogues Corpus and is able to output a command specific to the IT help desk context. In fact, our model has successfully learned how to use command line arguments, emojis, and other

Input Statement	Baseline Model	Advanced Model
How do I install Ubuntu on my computer?	sudo apt-get install ubuntu-desktop sudo apt-get install openssh-server install apt-get install Ubuntu-server	sudo apt-get install ubuntu-desktop sudo apt-get install openssh-server install apt-get install Ubuntu-server

elements specific to the online help desk setup. Further notice that since the response is not generic, adding mutual information re-ranking does not distort the top outcomes here.

Input	Baseline Model	Advanced Model
I want to play mp3 flash and wmw/avi	what is the problem? what ' the problem? what version the problem?	what kind the problem? install is the problem? _unk is the problem? mplayer is the problem?

One general observation that we see within our testing set is that when presented with a statement as opposed to a question, the Seq2Seq model is more likely to output a generic response, such as *"what is the problem?*". We believe that this phenomenon is due to the inherent bias within our training set, which includes many question-answer pairs.

One such example of this behavior is presented in the table above; the baseline model surfaces the over-generic default response of *"what is the problem?"*, while the advanced model with re-ranking offers more interesting responses, such as *"mplayer is the problem?"*. Notice that the improvement is small, since the top 50 responses from the Seq2Seq models are syntactically similar, and the majority of responses contain the phrase *"be the problem"*. Nonetheless, our re-ranking is able to prioritize the association between "mp3", "flash", and "wmw/avi" with "mplayer" and increases its ranking in the list. Were we to increase the beam size and enlarge our search space, we may be able to find even more relevant and interesting results.

Still, the output from the advanced model is not perfect, as shown by outputs such as "what kind the problem?". Since we incorporate an anti-language model $-P(r)$ in our objective function, certain ungrammatical sentences also obtain a boost in MI scores. This suggests that in future iterations, we may want to incorporate the window function $-f(P(r))$ or use a more sophisticated language model to filter out extreme language scores and less coherent sentences from our search space. Alternatively, improving the original model to output more grammatical responses can also prune the search space, and therefore improve the final outputs.

5.3 Quantitative Analysis (Lack of a good measurement)

Throughout the project, we struggled to find an appropriate quantitative measurement to evaluate our models. Though the BLEU score is often used for Seq2Seq models such as Neural Machine Translation models, it is not necessarily a good statistic for our model. Since the possible response space is enormous, arguably there is no categorically right or categorically wrong response in conversations. We resort to the rather inefficient manual scoring and qualitative analysis approach due to the lack of alternatives. Had time allowed, we would have spent more time searching for and defining a better quality measurement to evaluate our models.

6 Conclusion

As shown, we can achieve reasonable performance using a neural dialogue generation model, and have improved upon this model using the beam search over a mutual information objective function. Still, there is significant room for improvement within both of our models. Both models still, to some extent, suffer from the "I don't know" problem. We may be able to resolve this by increasing the beam size to increase the search space for the mutual information re-ranking. Using a more complex language model could also improve the mutual information scoring function.

The Ubuntu Dialogue Corpus itself may be a source of this problem, as dialogues present in the corpus are often not representative of "real-life" conversations between individuals. Using a different dialogue corpus for training could reveal other complexities in our model to address.

In the future, we may experiment with using pre-trained word vectors, such as using word2vec vectors trained on the Google News corpus. This may help with capturing semantic information within dialogues. We may also experiment with a hierarchical structure to include semantic information over multiple dialogues, as done by Serban et. al [7]. This could allow the model to better keep track of information across multiple dialogues.

7 Contributions

For the baseline model, Harrison wrote and edited scripts and ran experiments for the models, while Chenye designed the structure of the models and looked up relevant papers online. For the advanced model, Harrison wrote and edited the codes for the language model, while Chenye implemented the beam search and incorporated the two RNNs to generate the final results. We would like to thank the CS224n professors and TA's for providing guidance throughout the process of design and implementation.

8 References

References

[1] Sequence-to-sequence models. `https://www.tensorflow.org/tutorials/seq2seq`. Accessed: 2017-03-21.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[3] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. *CoRR*, abs/1510.03055, 2015.

[4] Jiwei Li, Alexander H. Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. Dialogue learning with human-in-the-loop. *CoRR*, abs/1611.09823, 2016.

[5] Jiwei Li, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. *CoRR*, abs/1701.06547, 2017.

[6] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *CoRR*, abs/1506.08909, 2015.

[7] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. Hierarchical neural network generative models for movie dialogues. *CoRR*, abs/1507.04808, 2015.

[8] Oriol Vinyals and Quoc V. Le. A neural conversational model. *CoRR*, abs/1506.05869, 2015.

[9] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *CoRR*, abs/1409.2329, 2014.