Survey on the diversity problem in deep learning dialogue systems

Vincent Espanol, Audrey-Anne Guindon, Justine Massicotte and Jonathan Moatti
Department of Business Intelligence, HEC Montreal
{vincent.espanol, audrey-anne.guindon, justine.massicotte, jonathan.moatti}@hec.ca

Abstract

Introducing diversity in generated responses while maintaining coherence is a core task of dialogue systems (chabots). While some models have become increasingly accurate in generating conversations, they often result in trivial or redundant responses (e.g. "I don't know", "maybe"). In this paper, we present a survey on the diversity problem in dialogue systems. We focus on the problems of lack of variability, improper objective, and weak conditional signal. We implement a deterministic Seq2Seq dialogue model and a variational model (VED) with and without global attention. We further implement and test four decoding methods to improve diversity. We conclude that methods proposed to address the diversity problem can indeed help improve diversity in dialogue systems, but at the cost of coherence.

1 Introduction

Dialogue systems (chatbots) have become increasingly used as a way to provide assistance to users. Today, users not only look for relevant answers to their questions, but also expect to communicate through natural language. In recent years, deep learning has proved to be effective for conversational modelling and natural language generation tasks. While earlier versions of chatbots were designed to answer questions based on pre-fixed rules, which provided little flexibility, deep learning has quickly replaced these models with end-to-end trainable neural networks, allowing for more flexibility and generating more human-like interactions (Vinyals & Le, 2015; Li et al., 2015; Serban et al., 2016; Zhao et al., 2017). Currently, sequence-to-sequence learning (Seq2Seq), which is composed of an encoder-decoder structure using recurrent neural networks (RNN), holds the state-of-the-art performance for dialogue systems (Mnasri, 2019). However, dialogue responses generated by Seq2Seq models tend to have low diversity. Seq2Seq tend to produce highly generic responses (e.g., "I don't know") rather than meaningful answers (Li et al., 2015).

In recent years, there have been several types of approaches to diagnosing and addressing the low-diversity problem in dialogue systems. Variational encoder-decoders (VED) have been proposed to increase the diversity of generated sentences (Serban et al., 2016; Zhao et al., 2017; Bahuleyan et al., 2018). Attention mechanisms have also proved useful in improving the performance of Seq2Seq dialog generation tasks (Yao et al., 2016; Mei et al., 2017). Furthermore, top quality text generation has been achieved using models that rely on the randomness in the decoding method (Fan et al., 2018; Holtzman et al., 2018; Radford et al., 2019). Given these findings, there is interest in exploring in what ways these methods have improved diversity in response generation.

In the following paper, our aim is to test various methods for increasing the diversity of generated sentences in chatbots. The purpose of this paper is to understand the low-diversity problem, to understand what solutions have been proposed so far, and to explore possible

new approaches. In Section 2, we review the related literature. In Section 3, we describe in detail the models considered. In Section 4, we discuss the implementation of the experiments. In Section 5, we highlight the experiments' results. Finally, we discuss our findings in Section 6.

2 Related Works

The low-diversity problem in dialogue generation tasks has gotten a lot of attention in recent years and many methods have been proposed to address it. The dominant viewpoints on the low-diversity problem in the literature include: lack of variability, improper objective function, and weak conditional signal (Jiang and Rijke, 2018).

Serban et al. (2017) and Zhao et al. (2017) trace the cause of the low-diversity problem in Seq2Seq models back to the lack of model variability in deterministic models. To increase variability, Zhao et al. (2017) propose to introduce variational autoencoders (VAEs) to Seq2Seq models. Such architecture were further extended to a variational encoder-decoder (VED) to transform one sequence into another with variational properties (Serban et al., 2017; Zhou and Neubig, 2017; Bahuleyan et al., 2018). These methods help introduce stochasticity in responses without resorting to sampling from the decoder.

Li et al. (2015) has noticed that the MAP objective function may be the cause of the low-diversity problem, since it can favor certain responses by only maximizing $p(\mathbf{y}|\mathbf{x})$. Recent works have proposed methods for diverse beam search, using a task-specific diversity scoring function (Vijayakumar et al., 2018; Kulikov et al., 2019; Pal et al., 2006). However, as noted in Holtzman et al. (2020), while such technique encourage desirable properties in generation, they do not remove the need to choose an appropriate decoding method. Recently, more attention has been given to moving away from beam search and greedy search decoders towards more randomized sampling methods as way to increase diversity.

Finally, Tao et al. (2018) have suggested that the way in which the original attention signal focuses on particular parts of the input sequence is not strong enough for the Seq2Seq model to generate specific responses, thus causing the low-diversity problem. Global attention (Luong et al., 2015) has shown to provide better outputs by focusing the input globally rather than step by step. Additionally, Bahuleyan et al. (2018) demonstrate that by using a variational attention mechanism for VED we can avoid the "bypassing" phenomenon that has been observed when deterministic attention mechanisms are combined with VEDs.

3 Method

The brain of our chatbot is a sequence-to-sequence (Seq2Seq) model. The goal of a Seq2Seq model is to take a variable-length sequence as an input, and return a variable-length sequence as an output using a fixed-sized model. The following section describes our Seq2Seq model along with the attention used, the variational encoder-decoder framework, the loss function and the decoding methods considered.

3.1 Sequence-to-Sequence Response Generation

In the Encoder–Decoder framework, an encoder reads the input sentence, a sequence of vectors $\mathbf{x} = (x_1, \dots, x_T)$, into a vector $\boldsymbol{c}$. The most common approach is to use an RNN such that:

$$\boldsymbol{h}_t = f_\theta(x_t + \boldsymbol{h}_{t-1})$$

and

$$\boldsymbol{c} = q(\{\boldsymbol{h}_1, \boldsymbol{h}_2, \dots, \boldsymbol{h}_T\})$$

where $\boldsymbol{h}_t$ is a hidden state at a given time step, and $\boldsymbol{c}$ is a vector generated from the sequence

of the hidden states. *f* and *q* are nonlinear functions.

The decoder is trained to predict the next word $y_{t'}$ given the context vector $\boldsymbol{c}$ and all the previously predicted words $\{y_1, \ldots, y_{t'-1}\}$. In other words, the decoder defines a probability over the response **y** by decomposing the joint probability into the ordered conditionals:

$$p(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T} p(y_t|\{y_1, \ldots, y_{t-1}\}, \boldsymbol{c})$$

where $\mathbf{y} = (y_1, \ldots, y_{T'})$. With an RNN, each conditional probability is modeled as:

$$\boldsymbol{s}_t = p(y_t|\{y_1, \ldots, y_{t-1}\}, \boldsymbol{c}) = g(y_{t-1}, \boldsymbol{s}_{t-1}, \boldsymbol{c})$$

where *g* is a nonlinear, multi-layered function that outputs the probability of y_t, and $\boldsymbol{s}_t$ is the hidden state of the decoder RNN. The output probability distribution $\boldsymbol{o}_t$ over the vocabulary corpus at time step *t* can then be calculated as:

$$\boldsymbol{o}_t = \text{softmax}(y_{t-1}, \boldsymbol{s}_t)$$

The two components of the Encoder-Decoder are jointly trained to minimize the negative log-likelihood:

$$\min_{\boldsymbol{\theta}} -\frac{1}{n}\sum_{n=1}^{N} \log_{p_{\boldsymbol{\theta}}}(\mathbf{y}_n \mid \mathbf{x}_n)$$

where $\boldsymbol{\theta}$ is the set of the model parameters and each $(\mathbf{x}_n|\, \mathbf{y}_n)$ is an (input sequence, output sequence) pair from the training set.

Instead of using a standard RNN as formulated above, our model uses a multi-layered Gated Recurrent Unit (GRU) from Cho et al. (2014) for both the encoder and decoder. Specifically, we used a bidirectional variant of the GRU, meaning that we have two independent RNNs: one that is fed the input sequence in normal sequential order, and one that is fed the input sequence in reverse order. The advantage of using a bidirectional GRU instead of a standard RNN is that it encodes both past and future context.

3.2 Global Attention

A common problem with a vanilla Seq2Seq decoder is that relying solely on the context vector to encode the entire input sequence results is likely to result in information loss. To address this problem, attention mechanisms have been proposed to dynamically align $\mathbf{y} = (y_1, \ldots, y_{|\boldsymbol{y}|})$ and $\mathbf{x} = (x_1, \ldots, x_{|\boldsymbol{x}|})$ during generation.

Luong et al. improved upon earlier works on attention by creating global attention. The idea of global attention is to consider all the hidden states of the encoder when deriving the context vector $\boldsymbol{c}$ instead of proceeding by state. In this model, at each time step t, the model infers a variable-length alignment weight vector $\boldsymbol{\alpha}_t$ based on the current target step $\boldsymbol{s}_t$ and all source states $\boldsymbol{h}_x$:

$$\boldsymbol{\alpha}_t = \frac{\exp(\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_x))}{\sum_{t'=1}^{|\boldsymbol{x}|} \exp(\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_{t'}))}$$

Here, score refers to a content-based function for which we considered the dot product of $\boldsymbol{s}_t$ and $\boldsymbol{h}_x$. The global context vector is then computed as the weighted average, according to $\boldsymbol{\alpha}_t$, over all the source hidden states:

$$\boldsymbol{\alpha} = \sum_{t=1}^{|x|} \boldsymbol{\alpha}_t \, \boldsymbol{h}_t$$

which is then fed back to the decoder.

3.3 Variational Autoencoders

In dialogue tasks, we would like to transform the source information into target information. A variational encoder-decoder (VED) framework transforms an input $\mathbf{x}$ to an output $\mathbf{y}$. Compared to a vanilla Seq2Seq model, here we add a latent variable z along with the context vector $\boldsymbol{c}$ and the target response $\mathbf{y}$. The task is then to model the true probability of a response $\mathbf{y}$ given an input $\mathbf{x}$. To do this, we introduce the latent variable z with a standard Gaussian prior $P(z) = \mathcal{N}(0, I_n)$ and factor $p(\mathbf{y}|\mathbf{x})$ using the probability density function:

$$p(\mathbf{y}|\mathbf{x}) = \int_z p(\mathbf{y}|z, \mathbf{x}) p(z) dz$$

At training time, we follow the variational autoencoder framework (Kingma and Welling, 2014), and approximate the posterior $p(z|\mathbf{x}, \mathbf{y})$. At test time, we sample a latent variable z and generate $\mathbf{y}$ through the response decoder $p(z|\mathbf{x}, \mathbf{y})$.

The resulting model proceeds as follows. The model encodes a tokenized input sentence and returns the hidden state $\boldsymbol{h}_x$ and the context vector $\boldsymbol{c}$. The hidden state $\boldsymbol{h}_x$ is concatenated with the hidden state from the target output $\boldsymbol{h}_y$ and passed to the VED model where we calculate the mean and variance of $p(z|\mathbf{x}, \mathbf{y})$ according to the method in Cao and Clark (2017). We then concatenate z and $\boldsymbol{h}_x$ and z and $\boldsymbol{c}$ and feed both along with the initial hidden state $\boldsymbol{s}_0$ into the decoder. The decoder feeds the current hidden state $\boldsymbol{s}_t$ and the encoder output into the global attention layer, which provides a set of attention weights that are summed to obtain the attention vector $\boldsymbol{\alpha}$, which corresponds to our new context vector. The new context vector is concatenated with the decoder output before passing through a linear layer. The probability over the corpus is then obtained using the softmax function.

3.4 Objective Function

The objective function for our deterministic models (vanilla Seq2Seq and Seq2Seq with attention) attempts to minimize the negative log likelihood. In the VED framework, we approximate the posterior $p(z|\mathbf{x}, \mathbf{y})$ with a proposal distribution $q(z|\mathbf{x}, \mathbf{y})$, which, like Cao and Clark (2017), we have chosen to be a diagonal Gaussian whose parameters depend on $\mathbf{x}$ and $\mathbf{y}$. Thus, we obtain the likelihood for this model using the Kullback–Leibler divergence (KL loss):

$$\mathbb{E}_{z\sim q} \log\left(p(\mathbf{y}|z, \mathbf{x})\right) - \mathrm{KL}(q(z|\mathbf{x}, \mathbf{y})||p(z))$$

We then combine the negative log likelihood with the KL loss to obtain the objective function for the VED model.

3.5 Decoding Strategies

In order to obtain the generated output sequence there are several techniques that can be used to decode from the probabilities of the words. Various decoding strategies can be used for word selection. For the purpose of our research, we use and compare different inference methods with the goal of improving diversity.

3.5.1 Maximization-based decoding

The most commonly used decoding objective is maximization-based decoding. Based on the

assumption that the model assigns a higher probability to higher quality outputs, these decoding methods search for the output token with the highest likelihood. A greedy search decoder (our baseline) simply chooses the word from the token with the highest softmax value. While this decoding method is optimal over a single time-step, the optimal argmax sequence is not tractable over multiple time steps (Chen et al., 2018). A common practice to deal with this problem is to use beam search (Li et al., 2016; Shen et al., 2017). Beam search involves selecting the top scoring k-words based on conditional probability at each step rather than simply maximizing the probability of a sequence of words, which can promote diversity by exploring different alternative. It has been observed, however, that diversity of responses is highly dependent on the selected beam size.

3.5.2 Top-k Sampling

Top-k sampling has recently become a popular alternative sampling procedure (Fan et al., 2018; Holtzman et al., 2018; Radford et al., 2019). At each time step, the top k next tokens are sampled from the distribution according to their likelihood. Formally, given a distribution $P(y_t|y_{1:t-1})$, we define its top-k vocabulary V^k as the set of size k which maximizes $\sum_{y\in V^k} P(y_t|y_{1:t-1})$. We then randomly select a value belonging to V^k that corresponds to the next word token in the output sentence.

3.5.3 Nucleus Sampling

Nucleus sampling was introduced by Holtzman et al. (2020) as a way to use the shape of the probability distribution to determine the set of tokens to be sampled from. In practice this means selecting the highest probability tokens whose cumulative probability mass exceeds the pre-chosen threshold of p. The size of the sampling set will adjust dynamically based on the shape of the probability distribution at each time step. For high values of p, this is a small subset of vocabulary that takes up vast majority of the probability mass, which the authors call the nucleus. Nucleus sampling can also be combined with top-k sampling to give Top-k Top-p sampling, which involves first limiting the distribution using k and then proceeding with Nucleus sampling using p.

4 Experiments

4.1 Dataset and Evaluation Metrics

We used the Cornell Movie-Dialogs Corpus as our dataset, which contains 220,579 conversational exchanges between 10,292 pairs of movie characters. We trimmed the dataset down to 90,000 sentence pairs of dialogue. We then randomly split the data into two sets of 45,000 pairs of dialog for the training and test set respectively. To evaluate the fluency and the diversity of the results, we used different evaluation metrics. BLEU calculates the percentage of n-gram matching between all of the generated sentences and all of the reference sentences (Papineni et al. 2002). We calculated the sentence-level BLEU-1 and BLEU-2 scores that measure the degree of unigram and bigram matching respectively. However, since it is debated how well this automatic metric is correlated with true response quality (Liu et al., 2016) and it does not measure the diversity of responses, we also used two other evaluation metrics. Distinct-N is a metric that focuses on the number of distinct n-gram of a sentence and was used to measure the diversity of a sentence (Li et al. 2016). We calculated the DIST-2 score that measures the degree of bigram diversity. In order to assess the quality of the text in terms of fluency, we also picked output examples from interactions with our chatbot and conducted a human evaluation.

4.2 Training details

We used a bidirectional variant of the GRU as our RNNs with 500 hidden units and two layers for both the encoder and decoder; the dimension of the latent vector z was also 500d.

We initially planned to use BERT fine-tuned on SQuAD as our pre-trained embeddings, but found it confounded the results and led to a decrease in performance. Therefore, we trained the embedding layer using the word indices. For both the target and the source text, the vocabulary was limited to the most frequent 7,823 tokens. We used the Adam optimizer to train all models, with a learning rate of 0.0001. To facilitate training two optimization tricks were used. The first trick we used was the teacher forcing algorithm. The second trick we used was gradient clipping with a clip equal to 50. While training the VED model, we noted the same difficulties as Bowman et al. (2016), as the model ignores the latent variable. We overcome this by gradually annealing the KL term weight over the course of training and using word dropout with a dropout rate of 0.1. All hyperparameter tuning was based on performance on the vanilla Seq2Seq model with global attention. Hyperparameter tuning was conducted in the same way to determine the beam size, top-k, and top-p values for the decoders. The same hyperparameters were used for all models described in Section 3.

5 Results

Table 1 represents the overall performance of various models. The generated responses can be viewed in Appendix A. We first implemented a vanilla Seq2Seq model, which we call our deterministic model. We then incorporated global attention in this model, which did not significantly improve BLEU or Distinct-N, although a slight improvement can be observed. We can more clearly see the improvement when we look at the generated outputs. Especially for Nucleus sampling, which tended to struggle with irrelevance, we found attention improved relevance to the input and provided a more coherent response. For the deterministic model, we report results obtained by using the decoding algorithms: greedy search, beam search, top-k sampling, and Nucleus sampling (Holtzman et al., 2020). In terms of diversity, top-k performed the best for the Distinct-N metric, but this was paired with a low BLEU score. Unsurprisingly, greedy search and beam search performed slightly better in terms of fluency as measured by BLEU-1 and BLEU-2, but worse in terms of diversity as measured by Dist-2. Overall, it seems that attention offers a slight improvement in terms of fluency. Based on the metrics, beam search appeared to offer the best balance between BLEU-1, BLEU-2 and Dist-2. However, as explained in the following subsection, based on human evaluation Nucleus decoding and VED provided the best balance between language fluency and diversity.

Model	**BLEU-1**	**BLEU-2**	**Dist-2**
Deterministic (no attn) + Greedy Search	0.153	0.068	0.712
Deterministic (no attn) + Beam Search, $b = 3$	0.147	0.061	0.800
Deterministic (no attn) + Top-k, $k = 5$	0.108	0.054	**0.891**
Deterministic (no attn) + Nucleus, $p = 0.9$	0.121	0.048	0.733
Deterministic (Luong attn) + Greedy Search	0.152	**0.069**	0.715
Deterministic (Luong attn) + Beam Search	**0.155**	**0.069**	0.737
Deterministic (Luong attn) + Top-k	0.108	0.054	**0.892**
Deterministic (Luong attn) + Nucleus	0.147	0.066	0.736
Deterministic (Luong attn) + Top-k Nucleus	0.136	0.044	0.722
VED (no attn)	0.124	0.049	0.742
VED (Luong attn)	0.124	0.050	0.741

Table 1: The results of BLEU and Distinct-N scores.

5.1 Human Evaluation

In order to assess the quality of the generated dialogue, we also conducted a human evaluation of the outputs. We randomly selected 50 questions and evaluated the outputs based on coherence and

relevance to the question – a subset of those outputs is presented in Appendix A. As expected greedy search and beam search provided the most coherent outputs, but resulted in low diversity, and repetitiveness ("I don't know"). In contrast, top-k sampling achieved high diversity outputs, but low language fluency. Nucleus sampling managed to generate more diverse sentences with high language fluency; however, the dialogue generated was often unrelated to the input sentence. The reason for this appeared to be the large sample space of candidate output words at the beginning of the selection. To decrease the initial selection, we used a combination of top-k and Nucleus sampling, which yielded a slightly higher diversity while preserving high language fluency and relevance to the input. Finally, VED managed to increase diversity, while preserving fluency, but suffered the same problem as Nucleus sampling and did not yield answers pertinent to the input. Attention brought improvement to VED in terms of relevance.

6 Discussion

We observed a clear tradeoff between repetition of key outputs ("I don't know"), incoherence of outputs ("I don t t never knew he"), and irrelevance to the input in response generation ("Look a cloud! I love clouds."). Repetition occurred when the word token distribution was peaked, concentrating the probability mass into just a few tokens. In contrast, incoherence occurred when a flat distribution led to many moderately probable tokens. Based on our observations, irrelevance to the input typically occurred in cases where the initial token was selected from a flat distribution, which became more peaked as selection proceeded, giving a heavier weight to the tokens generated in the previous time step than the input dialogue context.

As postulated in Holtzman et al. (2020), human language does not appear to maximize probability. Their research shows that the per-token probability of natural text is much lower on average than text generated using beam search, and that natural language rarely remains in a high probability zone for multiple consecutive time steps, instead veering into lower-probability but more informative word tokens. Our observations agreed with their results. Given this problem, it seems unlikely that high accuracy, high diversity sentences can be generated using beam search or greedy search, as they tend to remain within a high probability token space.

Ideally, the model would be able to sample from low probability tokens with high relevance to the input sequence. For example, if a question is about health, the health related tokens would have a higher probability despite holding a low probability over the entire corpus. This phenomenon can help explain the success of personality based and goal based chatbots in overcoming the tradeoff between diversity, fluency, and relevance.

Considering the problem at hand, the VED approach is attractive because it attempts to model the response probability along with the response generation. Unlike translation, which can proceed locally, dialogue needs to proceed globally. The VED-based approach bypasses the low-diversity problem by introducing randomness through the latent variable. Bahuleyan et al. (2018) observed a bypassing phenomenon in VEA/VED, where a VED with deterministic attention might learn reconstruction mostly from attention and propose variational attention to alleviate the problem.

The challenge observed was in maintaining the relationship between the input and output sentence while preserving language fluency and promoting high-diversity sentences. While multi-headed attention could improve relevance, a contributing factor to the problem appears to be with the objective and the data involved. The Cornell Movie-Dialogs Corpus is a popular dataset for text generation as it contains high-diversity sentences with clear pairs of dialogues. However, movie dialogs do not represent real, natural conversations since they represent fictional situations and often the conversations themselves are goal-oriented, involving outside factors related to the current scene in the movie. As a result, it is likely that the context vector was not able to meaningfully relate the input context with the output utterance. In future, works other datasets should be considered and potentially combined to balance dialogue diversity with contextually relevant inputs and output responses.

7 Conclusion

In this paper, we tested various methods for increasing the diversity of generated sentences in dialogue systems (chatbots). We implemented a deterministic model and a variational model both with and without global attention. We further implemented five different decoding methods on our deterministic model, namely: greedy search, beam search, top-k sampling, Nucleus sampling, and Top-k Top-p sampling. We investigate the diversity of output generated from these models. We show that decoding methods such as top-k and Nucleus sampling help improve response diversity, at the cost of coherence. We show a similar effect with the VED variational model. Overall, it appears that a combination of approaches: improved model variability, objective function, and attention will be needed to deal with the diversity problem. In future works, we would like to further explore the problem of relevance to input sequences observed in high fluency, high diversity responses with VED and the Nucleus decoder.

References

[1] Bahdanau, D., Bengio, Y., Bougares, D., Cho, K., Gulcehre, C., Schwenk, H. & Van Merrienboer, B. V. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the Empirical Methods in Natural Language Processing*.

[2] Bahuleyab, H., Mou, L., Poupart, P. & Vechtomova, O. (2018). Variational Attention for Sequence-to-Sequence Models. In *COLING*.

[3] Cao, K. & Clark, S. (2017). Latent Variable Dialogue Models and Their Diversity. In In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pp.182–187.

[4] Csaky, R. (2019). Deep learning based chatbot models. In *Technical Report*, pp.1-67.

[5] Jiang, S. & de Rijke, M. (2018). Why are Sequence-to-Sequence Models So Dull? Understanding the Low-Diversity Problem of Chatbots. In *Proceedings of the 2018 EMNLP Workshop SCAI*.

[6] Jurafsky, D., Li, J. Monroe, W. & Ritter, A. (2016). Deep Reinforcement Learning for Dialogue Generation. In *EMNLP*.

[7] Holtzman, A., Buys, J., Du, L., Forbes, M., Choi, Y. (2020). The Curious Case of Neural Text DeGeneration.In *ICLR*.

[8] Le, H., Nguyen, T., Tran, T. & Venkatesh, S. (2018). Variational Memory Encoder-Decoder. In *Advances in Neural Information Processing Systems*, pp.1515–1525.

[9] Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, B. (2015). A Diversity Promoting Objective Function for Neural Conversation Models.

[10] Liu, C.-W., Lowe, R., Serban, I. V., Noseworthy, M., Charlin, L. & Pineau, J. (2016). How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation. In *EMNLP*.

[11] Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. In *ICLR Workshop Papers*.

[12] Mnasri, M. (2019). Recent Advances in Conversational NLP : Towards the Standardization of Chatbot Building.

[13] Tao, C., Gao, S., Shang, M., Wu, W., Zhao, D. & Yan, R. (2018). Get The Point of My Utterance! Learning Towards Effective Responses with Multi-Head Attention Mechanism. In *IJCAI*, pp.4418–4424.

[14] Tong, X., Li, Y. & Yeng, C.-M. (2017). Variational Neural Conversational Model. In *ICML Deep Learning Workshop*.

[15] Serban, I. V., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A. C. & Bengio, Y. (2017). A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues. In *AAAI*, pp.3295–3301.

[16] Shao, L., Gouws, S., Britz, D., Goldie, A., Strope, B. & Kurzweil, R. (2017). Generating High-Quality and Informative Conversation Responses with Sequence-to-Sequence Models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pp.2210–2219.

[17] Vijayakumar, A. K., Cogswell, M., Selvaraju, R. R., Sun, Q., Lee, S., Crandall, D. and Batra, D. (2016). Diverse beam search: Decoding diverse solutions from neural sequence models.

[18] Vinyals, O. & Le, Q. (2015). A Neural Conversational Model.

[19] Zhao, T., Zhao, R. & Eskenazi, M. (2017). Learning Discourse-level Diversity for Neural Dialog Models using Conditional Variational Autoencoders. In *ACL*, pp.654-664

A Responses

No Attention + Greedy

```
> Hello
Bot: hello .
> How is it going ?
Bot: i don t know .
> What is your favorite song ?
Bot: i don t know .
> How old are you ?
Bot: i don t know .
> Where are you from ?
Bot: i m in the room .
> I love you .
Bot: i love you too .
> Goodbye !
Bot: i m not !
> q
```

No Attention + Beam

```
> Hello
Bot: how do you know who you are ?
> How is it going ?
Bot: i don t know .
> What is your favorite song ?
Bot: i don t know .
> How old are you ?
Bot: i don t know .
> Where are you from ?
Bot: i don t know .
> I love you .
Bot: i love you too .
> Goodbye !
Bot: all right . . .
> q
```

No Attention + Top-K

```
> Hello
Bot: who else has been by the toilet parts or anything
> How is it going ?
Bot: okay . fine thank your eyes sir it was me
> What is your favorite song ?
Bot: she called your mother s dead . . .busy
> How old are you ?
Bot: i told . we are things friends today for him
> Where are you from ?
Bot: the united boys .a street and ? is !
> I love you .
Bot: yeah i can do it back in now captain .
> Goodbye !
Bot: don t ! ! fuck god i am calm down
> q
```

No Attention + Nucleus

```
> Hello
Bot: how are you crabtree ?
> How is it going ?
Bot: you told me you were right
> What is your favorite song ?
Bot: i got a dream about it .
> How old are you ?
Bot: he must have time !
> Where are you from ?
Bot: last night .
> I love you .
Bot: you and walks ?
> Goodbye !
Bot: you know what you mean .
> q
```

Attention + Greedy

```
> Hello
Bot: hello .
> How is it going ?
Bot: i don t know .
> What is your favorite song ?
Bot: i don t know .
> How old are you ?
Bot: i m fine .
> Where are you from ?
Bot: southern .
> I love you .
Bot: i love you .
> Goodbye !
Bot: . . .
> q
```

Attention + Beam

```
> Hello
Bot: hello .
> How is it going ?
Bot: i don t know .
> What is your favorite song ?
Bot: i don t know .
> How old are you ?
Bot: i don t know .
> Where are you from ?
Bot: san francisco .
> I love you .
Bot: i love you too .
> Goodbye !
Bot: . . .i can t . . .
> q
```

Attention + Top-K

```
> Hello
Bot: you . what you say ! we can do !
> How is it going ?
Bot: fine fine okay . how you doin tonight !
> What is your favorite song ?
Bot: why what s your apartment little relationship ?
> How old are you ?
Bot: four with four ! you re tootin s ! okay
> Where are you from ?
Bot: south city ! ! otis ? ? ! now what
> I love you .
Bot: . tell yourself what you think so wishes me it
> Goodbye !
Bot: . let him do you do for them s wife
> q
```

Attention + Nucleus

```
> Hello
Bot: hello .
> How are you doing?
Bot: i m fine .
> What is your favorite song?
Bot: i m not gonna marry him .
> How old are you?
Bot: i m not .
> Where are you from?
Bot: san francisco .
> i love you
Bot: i love you too .
> goodbye
Bot: no .
> q
```

Attention + Top-K + Nucleus

```
> Hello
Bot: hello .
> How is it going?
Bot: fine .
> What is your favorite song?
Bot: i m not sure .
> How old are you?
Bot: seventeen .
> Where are you from?
Bot: san francisco .
> i love you
Bot: i love you too .
> goodbye
Bot: how long ?
> q
```

VED + No Attention

```
> Hello
Bot: hi mary . how are you ?
> How is it going ?
Bot: it s delicious trust me .
> What is your favorite song ?
Bot: he s got a reason .
> How old are you ?
Bot: you re a little confused ?
> Where are you from ?
Bot: up north .
> I love you .
Bot: i love you .
> Goodbye !
Bot: what ? !
> q
```

VED + Attention

```
> Hello
Bot: hi . !
> How is it going ?
Bot: i don t know .
> What is your favorite song ?
Bot: nothing .
> How old are you ?
Bot: twenty five .
> Where are you from ?
Bot: san francisco .
> I love you .
Bot: you don t look it .
> Goodbye !
Bot: she s gonna kill you
> q
```

Human Responses

```
> Hello
Human: Hi!
> How is it going?
Human: I'm alright, how are you?
> What is your favorite song?
Human: Anything by the Beatles.
> How old are you?
Human: I'm 23.
> Where are you from?
Human: I'm from Toronto.
> I love you.
Human: I love you too.
> Goodbye!
Human: See ya.
```