Impact of Auxiliary Loss Functions on Dialogue Generation
Using Mutual Information

Jack St. Clair
Haverford College
370 Lancaster Ave
Haverford, Pennsylvania 19041
Email: jrstclair@haverford.edu

Thomas Conley and Jugal Kalita
University of Colorado Colorado Springs
1420 Austin Bluffs Pkwy
Colorado Springs, Colorado 80918
Email: tconley@uccs.edu, jkalita@uccs.edu

Abstract
Dialogue generation involves teaching a program to generate natural conversation. Assuming there are two participants, it requires developing a program that can converse with a human being or another program, and do so coherently and fluently. This paper presents the development of a dialog generating program, popularly called a chatbot, that learns from a corpus of conversations, using a basic sequence to sequence (Seq2Seq) model with a variety of auxiliary loss functions. Auxiliary loss functions are similar to loss functions used during training, but are instead used during generation and do not have to be differentiable. The auxiliary loss functions developed for this chatbot are variants of mutual information between the utterances of one speaker and those of the other, because the objective is to couple these utterances tightly. We demonstrate that using different forms of mutual information leads to developments of chatbots of varying quality. The research shows that when these different chatbots chat with themselves, it is not a sufficient replacement for a human.

Keywords: dialogue generation, Seq2Seq model, maximum mutual information

Introduction
To be able to participate in a dialog, or simply chat with others is a natural human ability. Certain people are good chatters and other people are drawn to them. In other words, such individuals are able to steer the conversation onto topics that are of interest to other participants, follow a topic of conversation for an extended duration, and add details as necessary. There are many books in the market that provide guidelines for interesting and engaging conversation, e.g., (Fine 2005) and (Wadsworth 2017). We focus here on what is called small talk in general parlance. Webster’s Dictionary defines small talk as “light or casual conversation: chitchat”.1 The Urban Dictionary defines small talk as “useless and unnecessary conversation attempted to fill the silence in an awkward situation”.2

Computer scientists have tried to build chatbots for a long time, starting from the initial attempt at building an artificial psycho-therapist called Eliza (Weizenbaum 1966). Because of the nature of psychotherapy, even with its limited abilities, Eliza was able to impress the populace at large, in addition to the research community. Eliza worked simply by pattern matching, and produced inane responses when pattern matching failed to produce a meaningful response. The frame-based architecture for conversation making, introduced by (Bobrow et al. 1977) in the GUS system, enconced itself as the predominant approach to building dialog agents for several decades. Apple’s SIRI and other digital assistants were built using this architecture (Bellegarda 2013; 2014; Jurafsky and Martin 2018). Such speech-based conversational agents used Partially Observable Markov Decision Process (Sondik 1971) in the context of the frame-based architecture, maintaining a system of beliefs and updating them using Bayesian inference. They also used reinforcement learning (Sutton and Barto 1998).

Recently, researchers had started building chatbots by training machine learning programs on transcripts of conversations. Ritter, Cherry, and Dolan (2011) presented a data-driven approach to generating responses to Twitter status posts, using statistical machine translation, treating a status post as a question and the response as its “translation”. Of late, researchers have built chatbots using Artificial Neural Networks or Deep Learning. Such research usually uses Seq2Seq models (Cho et al. 2014; Sutskever, Vinyals, and Le 2014). Seq2Seq models have been used by many recent chatbots (Vinyals and Le 2015; Li et al. 2016b; 2016a; Shao et al. 2017; Wu, Martinez, and Klyen 2018). Although the Seq2Seq framework has shown good results in dialogue generation, we believe that the evaluation of the dialogues can be better measured. Most approaches are composed of two Recurrent Neural Networks (RNN) or Long Short-Term Memory (LSTM) units, with the first encoding from words into vectors. The second then decodes these vectors back into words to create the output. The chatbot can thus create a response relevant to the input.

The research presented in this paper aims to examine the role that the use of various auxiliary loss functions plays in the quality of dialog generated when trained on several conversational corpora. Our contribution lies in detailed analysis of the dialogs at various levels of granularity, using a number or metrics. We believe that this is the first time such detailed analysis of automatically generated dialogs has been carried out. We use a simple RNN model for training the conversational agents in small talk since our focus

---

1https://www.merriam-webster.com
2https://www.urbandictionary.com/
is more on the auxiliary loss functions. We believe that these loss functions are likely to behave in similar ways with other agent architectures as well.

Problem Statement
Consider a dialog with two participants $Q$ and $A$. $Q$ initiates the conversation with a question, statement or comment $q_1$, and $A$ follows with a response or a follow-up statement or comment $a_i$. Thus, a conversation is a sequence of textual elements

$$< < q_1, a_1 >, < q_2, a_2 >, \cdots, < q_i, a_i >, \cdots, < q_k, a_k > > .$$

In this paper, we discuss the development of a conversational agent that can be either $Q$ or $A$ or both. We develop this agent by training a machine learning model (Seq2Seq) on a corpus of dialog.

In other words, we give a training sequence $T$ of conversations, ideally between two agents, to a Seq2Seq learner that learns the association between $q_i$ and $a_i$. This is done by optimizing an Artificial Neural Network (ANN) model, so that given an unseen $q$, the model can generate an appropriate $a$ based on the learned associations. Once it has been trained, it presumably becomes a “competent” small-talk chatter. During testing, the learned chatbot is given a previously unseen sentence $q$ and it responds with a sentence $a$. The conversation may continue for a while, and ends when a conversation end indicator is produced.

Conversational ability acquired through training using a Seq2Seq neural model depends on the “loss function” used during training. But, it is also possible to manipulate a trained network’s outputs during usage or testing to produce a variety of outputs as seen in this paper. When producing output, the final sentences are generated by searching through a set of candidates, and some candidates may turn out to be more appropriate based on additional processing.

Related Work
Using Seq2Seq models for dialogue generation has become commonplace in recent years. Ritter, Cherry, and Dolan (2011) were the first to use a model used for Statistical Machine Translation (SMT) to generate responses to queries by training on a corpus of query-response pairs. Sordoni et al. (2015) improved Ritter et al.’s work by re-scoring the output of the SMT-based response generation system with a Seq2Seq model that took context into account.

Vinyls and Le (2015) used an RNN-based Seq2Seq model using the cross-entropy auxiliary loss function and a greedy search at the output end. Wen et al. (2015) used LSTMs for joint planning of sentences and surface realization by adding an extra cell to the standard LSTM architecture (Hochreiter and Schmidhuber 1997), and using the cross-entropy loss. They produced sentence variations by sampling from sentence candidates. Li et al. (2016a) used Maximum Mutual Information (MMI) as the objective function to produce diverse, interesting and appropriate responses. This objective function was not used in the training of the network, but to find the best among candidates produced by the model at the output end during generation of responses. Our paper is substantially inspired by this work.

Li et al. (2016b) applied deep reinforcement learning using policy gradient methods to punish sequences that displayed certain unwanted properties of conversation: lack of informativity, incoherence and responding inanely. Lack of informativity was measured in terms of high semantic similarity between consecutive turns of the same agent. Semantic coherence was measured in terms of mutual information, and low values were used to penalize ungrammatical or incoherent responses. The approach also gave negative rewards for inane responses that belong to one of a number of inane responses such as “I don’t know”.

Su et al. (2018) use a hierarchical multi-layered decoding network to generate complex sentences. The layers are GRU-based (Cho et al. 2014), and each layer generates words associated with a specific Part-Of-Speech (POS) set. In particular, the first layer of the decoder generates nouns and pronouns; the second layer generates verbs, the third layer adjectives and adverbs; and the fourth layer, words belonging to other POSes. They also use a technique called teacher forcing (Williams and Zipser 1989) to train RNNs using the output from the prior step as an input.

In spite of the complex approaches that are being proposed to generate text in the context of question-answering, dialog generation or otherwise, the evaluation of the dialogs have been primarily being in terms of the BLEU (Bilingual Evaluation Understudy) score (Papineni et al. 2002), a metric that was designed for evaluation of SMT. BLEU scores are highly correlated with human judgments of quality for SMT at the corpus level. BLEU computes scores for individual translated sentences by comparing overlaps in terms of n-grams with a set of good quality reference translations. Those scores are then averaged over the whole corpus to reach an estimate of the translation’s overall quality. It does not take into account intelligibility or grammatical correctness, and is not a good measure of translations of individual sentences. BLEU score shines as a metric for SMT, however we believe that using BLEU scores alone for evaluating dialogs is limiting. Li et al. (2016b) used two additional computable metrics: the length of the dialog generated, and diversity by calculating the number of distinct unigrams and bigrams. These two are good additions to the BLEU metric, but we believe that it can be further expanded. Coh-Metrix (Graesser et al. 2004) is a Web-based tool that analyzes texts on over 200 measures of cohesion, language, and readability. We use Coh-Metrix in the evaluation of dialogs in this paper to provide a rich understanding of their quality.

Loss Function
Our training model employs a softmax cross entropy calculation on the logits as provided by TensorFlow. We experimented with hinge and sigmoid cross-entropy functions as an alternative loss measurement during training. Results with these other training functions were inconclusive and since we had no logical reasoning for trying alternate losses without a ground truth for comparison, we leave this area of research for future work.
Instead, we concentrate on the auxiliary loss function needed during sentence generation. These functions operate on partially generated sequences of states in a beam search. The measure of loss when evaluating these solution states used to find consensus among a number of choices equal to the beam width. We tested extensively using a beam width of 2 using four auxiliary loss functions.

We begin with a test using NET loss; by using no loss function at all we predict subsequent characters using only the probabilities predicted by the network.

Another function which uses MMI measurements as loss is shown in equation 2 where $S$ represents the current set of states during sentence generation in the beam search; $T$ represents the set of possible next states. This function is modeled after work conducted by (Li et al. 2016a) and is shown in Equation 2.

$$
\hat{T}_{MMI} = \arg \max_T \{ \log p(T|S) - \lambda \log p(T) \} \tag{2}
$$

We further develop this MMI approach by including Entropy normalization. This approach is inspired by (Estévez et al. 2009) who used Normalized MMI for feature selection. We calculate entropy from predicted network probabilities as shown in equations 3 and 4.

$$
H_S = \sum_{t=0}^{|S|} -P(S_t) \times \log(P(S_t)) \tag{3}
$$

$$
H_T = \sum_{t=0}^{|T|} -P(T_t) \times \log(P(T_t)) \tag{4}
$$

The minimum of these values is used to normalize our MMI value as in Equation 5.

$$
\hat{T}_{NORM} = \frac{\hat{T}_{MMI}}{\min(H_S, H_T)} \tag{5}
$$

Finally we experiment with MMI Entropy normalization where entropy is not calculated but measured directly from the training corpus in terms of character frequencies. Optimizing based on this function should affect the uniqueness of generated sentences.

**Architecture**

The core of our model is a stack of dense layers comprised of gated recurrent unit (GRUs) cells. We performed tests on a configuration with 3 layers, each divided into 3 blocks, where each block contained 2048 GRUs. This architecture is based on a prior implementation available at on-line.

The GRU stack is initialized with the previous state ($s_{t-1}$) and the current character encoding ($x_t$) at each time step $t$ in the character sequence. The GRU output ($Y_t$) and the weights from the final stack layer ($W_t$) are combined with a bias ($b$) to produce logits at time $t$. We define logits as the raw output of the GRU stack which can be normalized and passed to a softmax function to produce probabilities. In this scheme, we update the logits by applying weights and biases from the last GRU layer as shown in Equation 6. The logits are then passed to a loss function for back propagation within the GRU stack. We do not limit or pad the length of the input sequence but perform back propagation through time (BBTT), relying on TensorFlow’s default truncated back-propagation capabilities.

$$
Logits = (Output \times Weights) + Biases \tag{6}
$$

Note that, output sequences ($y_0, ..., y_t$) are not generated during the training phase where only the logits are used for back-propagation. It is after training, during testing or dialog generation, that the logits are converted to probability using softmax. Finally, probabilities are converted to character sequences using a beam search.

Our beam search employs custom loss functions based on Maximum Mutual Information (MMI) as described in (Li et al. 2016b). We extend this concept to include entropy-normalized MMI, which has been used for feature selection by (Trinh et al. 2018), and is used in this research to select the optimal path in our beam search.

Figure 1 illustrates a single time-step $t$ in sequence processing by our recurrent neural network.

The model accepts a (one-hot) binary vector $X$ and a previous state vector, $S$, as inputs and produces a state vector, $S$ and a predicted probability distribution vector $P_t$, for the (one-hot) binary vector $Y_t$.

**Evaluation Metrics**

The responses of the chatbot are impacted by the auxiliary loss function used. That is why we vary the loss function to examine how the choices of auxiliary loss functions and datasets change the nature and quality of the generated conversation.

Evaluating the responses automatically is difficult because as of yet, there is no good and agreed-upon way to evaluate a responses created by a chatbot (Liu et al. 2016).

---

1https://github.com/pender/chatbot-rnn
Word-overlap metrics such as BLEU (Papineni et al. 2002), METEOR (Banerjee and Lavie 2005) and ROUGE (Lin 2004) have been commonly used in the past, but we believe that high word overlap between a question and a response does not always make for good conversation, although some overlap shows continuity of topic and thus, coherence. The metrics used here are from a rich discourse evaluation suite called Coh-Metrix (Graesser et al. 2004), as mentioned earlier. Coh-Metrix is an online tool with over 100 different metrics that determine semantic and syntactic features of a given text. These metrics help determine how different loss functions and datasets differ in the output that the model generates. Whether the learning regimen imposed by the loss function and the processing of the output candidates using the MMI criterion help generate syntactically, semantically and discourse-wise effective conversations will be measured by a choice of Coh-Metrix metrics. We use the following metrics in this paper, although all of Coh-Metrix metrics for the conversations are available in Supplementary Material.

- **Mean Words per Sentence:** This metric calculates the number of words in each sentence and then gives the mean of their lengths.

- **Narrativity:** This is a complex metric that measures the narrative or story telling qualities of a text. A narrative text has characters, places, events and chronology of events in it. Narrativity is higher on texts with reoccurring people, places and things. Novels and dramas are examples of narrative text, whereas informational texts are not unless they are written deliberately in a narrative manner. However, every text has some elements of narrativity in it, and Coh-Metrix combines 17 simple metrics and computes a single narrativity number (Graesser, McNamara, and Kulikowich 2011). Making a text narrative makes it easier to follow. Good conversation usually follows a narrative genre. Its value is in the range 0-100.

- **Syntactic Simplicity:** Texts with fewer words and simple sentence structures will receive high scores. Sentences with a lot of words and complex syntax will receive low scores. The value is between 0 and 100.

- **Referential Cohesion:** Texts with words that continue to be mentioned throughout the text receive high referential cohesion scores. This is a simple measure of cohesion that measures the overlap of nouns, pronouns, content words, etc., in adjacent sentences (question-answer pairs, in our case) as well as in the entire text. The value ranges from 0-100. Higher cohesion means the discourse is easier to follow.

- **Sentence Semantic Similarity:** This metric is calculated by latent semantic analysis and does this for all sentences. It looks at the meaning of each sentence and sees if there are any similar themes within adjacent sentences. The value is between 0 and 1. Latent Semantic Analysis (Deerwester et al. 1990; Landauer et al. 2013) can be used to measure semantic overlap among sentences and paragraphs. LSA creates word co-occurrence matrix for words in the document or a smaller unit of the document, performs matrix decomposition and size reduction to obtain representational vectors for individual words. Coh-Metrix computes 8 LSA metrics: LSA cosines between adjacent sentences (in our case, question-answer pairs), sentences in a paragraph, sentences in adjacent paragraphs, their means and standard deviations, etc. We present only one of these in our results.

- **Lexical Diversity:** Lexical diversity calculates the type-token ratio for all words in a text. Type-token ratio (TTR) is defined as “the number of unique words divided by the number of tokens of these words” (Templin 1957). Each unique word is called a type, and each occurrence a token. When TTR is around 1, each word occurs only once, making comprehension difficult since each word needs to be integrated with the conversation. When TTR is lower, words are repeated in the conversation, and it is easier to process. TTR is computed for content words only.

- **Connective Word Occurrence:** This metric calculates the diversity of words throughout the text, higher lexical diversity means a higher score.

- **Modifiers per Noun Phrase:** This metric calculates the number of modifiers in all noun phrases in the text and then takes the mean of those values.

- **Sentence Syntax Similarity:** This metric takes the syntax trees of all sentences in the text and compares them, calculating the number of similar nodes between the trees.

- **Content Word Frequency:** This metric calculates the average occurrence of content words. Content words include nouns that refer to objects of conversation, lexical or non-auxiliary verbs that describe what can be done with or to these objects, adjectives and adverbs describing qualities of the objects. Although there are only about 150 non-content of function words, they are used heavily in conversation or text. The high presence of content words in a conversation is likely to indicate that the conversation regards something substantial, rather that something meaningless.

- **Word Familiarity:** A piece of text is scored based on the average familiarity of all the words in it. The metric uses familiarity scores assigned to 3488 words in a database (Coltheart 1981) of words that were rated on a 7-point scale by adult raters, 1 being given to words previously unseen, and 7 to words that are seen almost daily. The ratings are multiplied by 100. Sentences with more familiar words can be processed and understood quickly. In a small-talk situation, high familiarity is important, but not so in formal or academic exchanges.

- **Reading Ease:** This metric provides the Flesch reading score for the entire text where a higher score means that the text is easier to read. The formula used is given below (Flesch 1948).

\[
  r = 206.835 - \left(1.015 \times \frac{sl}{spw}\right) - (84.6 \times \frac{syw}{spw})
\]

where \( r \) is Reading Ease, \( \frac{sl}{spw} \) is the average sentence length, and \( \frac{syw}{spw} \) is the average number of syllables per word. A Flesch score of 90-100 signifies that the text is...
at the 5th grade level, easily understood by a typical 11-year old. A score of 0-30 indicates readability at college graduate level, signifying high difficulty.

Experiments and Results

Our model is trained on data from multiple locations: over 2 GBs of conversations in the comments of Reddit posts, and data from proceedings in the Supreme Court of the United States (SCOTUS). The model is also trained on two other datasets; the Cornell movie corpus (Danescu-Niculescu-Mizil and Lee 2011), a corpus of the scripts from over 600 movies and also on Shakespeare’s Romeo and Juliet. These datasets are used for training and then by running the trained model, one can converse with the chatbot.

Metrics Results

Multiple tests were run using Coh-Metrix and the Reddit trained neural network as well as the four distinct auxiliary loss functions NET, MMI, NORM and ENT described in this research. All generated conversations consist of 15 question and answer pairs generated by two different chatbots. From this data, some interesting trends can be observed.

![Figure 2: Number of sentences as a measure of sophistication for 4 auxiliary loss functions.](image)

![Figure 3: Coh-Metrix results, the blue dotted line is Narrativity, the orange dashed line is Referential Cohesion and the green dotted and dashed line is Reading Ease.](image)

![Figure 4: Coh-Metrix results, the blue dotted line is Narrativity, the orange dashed line is Referential Cohesion and the green dotted and dashed line is Reading Ease.](image)

Parts (a) and (b) in Figure 2 show conversations between two chatbots where one is generated using MMI and the other is generated using NET. These figures may suggest that the information the MMI is based on, namely the semantics of the sentence could be harder to decipher using the MMI model instead of the simpler way that the NET model looks at the semantics. Due to the plethora of things that the MMI is calculating, the MMI predictions for characters may be more directed towards simplicity due to the overflow of information the generator is being sent. The simpler NET model may not be able to get as much information but because it gets less information it is able to create a constant dialogue between two chatbots that are simpler in their responses.

In contrast, looking at parts (c) and (d) in Figure 2, the number of sentences in the ENT and NORM generated dialogues are almost identical. The more complex NORM calculation is actually able to barely hold the conversation for longer than the simpler ENT model. NORM is almost the

---

Table 1: Coh-Metrix values for different generated texts

<table>
<thead>
<tr>
<th></th>
<th>NET</th>
<th>MMI</th>
<th>NORM</th>
<th>ENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Words per Sentence</td>
<td>10.079</td>
<td>3.200</td>
<td>1.550</td>
<td>51.389</td>
</tr>
<tr>
<td>Narrativity</td>
<td>99.910</td>
<td>98.170</td>
<td>57.140</td>
<td>78.810</td>
</tr>
<tr>
<td>Syntactic Simplicity</td>
<td>58.320</td>
<td>41.680</td>
<td>99.930</td>
<td>0.160</td>
</tr>
<tr>
<td>Referential Cohesion</td>
<td>90.820</td>
<td>64.800</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Sentence Semantic Similarity</td>
<td>0.363</td>
<td>0.359</td>
<td>0.167</td>
<td>0.624</td>
</tr>
<tr>
<td>Lexical Diversity</td>
<td>0.366</td>
<td>0.594</td>
<td>0.333</td>
<td>0.096</td>
</tr>
<tr>
<td>Connective Word Occurrence</td>
<td>48.499</td>
<td>0</td>
<td>0</td>
<td>57.297</td>
</tr>
<tr>
<td>Modifiers per Noun Phrase</td>
<td>0.408</td>
<td>0.231</td>
<td>0</td>
<td>0.908</td>
</tr>
<tr>
<td>Sentence Syntax Similarity</td>
<td>0.114</td>
<td>0.158</td>
<td>0.593</td>
<td>0.040</td>
</tr>
<tr>
<td>Content Word Frequency</td>
<td>2.813</td>
<td>4.580</td>
<td>2.358</td>
<td>2.835</td>
</tr>
<tr>
<td>Word Familiarity</td>
<td>589.115</td>
<td>572</td>
<td>591.5</td>
<td>583.183</td>
</tr>
<tr>
<td>Reading Ease</td>
<td>90.526</td>
<td>100</td>
<td>98.835</td>
<td>63.476</td>
</tr>
</tbody>
</table>

---

https://github.com/ravexina/shakespeare-plays-dataset-scaper
answers show greater similarity than random answers in three training corpus. This is shown to be true as generated system when compared with a random response from the Greater sentence similarity is expected from an intelligent question and answer with four commonly used metrics. The fact that the NORM model also has a 0.593 sentence syntax similarity, which is 0.435 more than the next model (MMI with a 0.158 sentence syntax similarity) shows the monotony of the NORM generated dialog. The same conclusion can be taken from the 0.167 value for NORM sentence semantic similarity, which is the lowest of all the models. This is interesting because MMI models were found to improve the output of chatbots in (Li et al. 2016b).

After more tests generating sets of 15 question-response pairs, the reasons for the failure of the NORM generator trained on Reddit data are revealed. On all other datasets, the two chatbots chatting are able to create 15 pairs of dialogue. The most probable reason for this is the huge disparity in the size of the datasets, the data other than Reddit were trained on 10,000 lines of dialogue (other than Romeo and Juliet which is only 840 lines). As can be seen in Figure 3, the Reddit dataset stands out as having low narrativity, but high reading ease. This happens because when it does not generate sentences, there is no story, yet the lack of sentences is easy to read. Figure 4 shows this same fact, only the NET model shows high narrativity and reading ease, it was also the only model to create the dialogue. It now seems obvious, that the less data that is present and the fewer hoops to jump through in generation, the fewer probabilities the generator has to consider. This could show that more data may only be better with chatbots conversing with humans, because when chatbots are chatting with each other, the dialogue needs to create long enough sentences to keep the flow of information. Thus the less information passed to each chatbot, and the more information they have in common, the easier it will be for them to converse. With human input however, more data is required because the human can ask anything and will expect a coherent response from the chatbot. This is only possible if the chatbot has seen data similar to the input, those chances are of course higher with more data.

Figure 5 shows the results of comparing similarity between network generated responses to randomly selected questions in the corpus. We measure similarity between question and answer with four commonly used metrics. Greater sentence similarity is expected from an intelligent system when compared with a random response from the training corpus. This is shown to be true as generated answers show greater similarity than random answers in three of the four selected metrics.

![Figure 5: Comparison of auxiliary loss functions by 4 string distance measurements including comparison with randomly selected questions and answers.](image)

**Generation Results**

Table 2 illustrates a number of things about generated dialogue. First of all, we suppose that the capital letters at the beginning of the sequence are the chatbot’s best attempt at writing the name of a supreme court justice as each sentence in the training corpus is labeled thus. The example in this example is most likely “JUSTICE SCALIA:” as represented in the 100 corpus records. While the poorly trained model did not generate the actual name of a justice we admire the attempt considering it is based on only 100 lines of data.

This dialogue in Table 2 is pretty good, but this trained model does not create very coherent dialogue between participants and it also often reaches the character limit for a response which makes the dialogue end abruptly. The chatbot trained on 10,000 lines of the Cornell movie corpus does not give very relevant answers and some of the answers themselves are hard to decipher:

“You know you’ve got to go to ask you telling you, I’m afraid of yourself.”

Although this is not a grammatical sentence we observe that it does seem to flow somewhat naturally. This begs the question, what is the best kind of dialogue to generate? Is meaning the most important part, or should it flow well?

**Conclusion and Future Work**

This paper took a Seq2Seq model chatbot and trained it on multiple datasets and used different auxiliary loss functions to see what would generate the best dialogue between two chatbots. Unexpectedly, the MMI and NORM auxiliary loss functions performed the poorest when trained on the largest dataset. This suggests that MMI based functions get confused with too much data, slowly degrading the dialogue. This concludes that trying to create a chatbot by testing it in a conversation with itself does not reveal whether or not a chatbot is intelligent.
Dialogue generation still has a long way to go in creating chatbots that can fluently converse in the same sophisticated natural language of humans. In future work, some easy ways to create better dialogue would be to add a reinforcement learning and attention model to the Seq2Seq model from this research. These models would help the chatbot create relevant responses. Other future additions that could help create a more comprehensive chatbot include some larger and more sophisticated datasets which would allow for more testing on the model and may create responses even more like that of humans than the implementation in this research. Furthermore, advancements in neural networks may bring about faster and more complex neural networks that could also help create even more human like responses. There is also potential for great strides in the ability of evaluation models. With better evaluation models, the dialogue generation model can learn more from itself. Additionally, as mentioned in (Wu, Martinez, and Klyen 2018), there is potential to differentiate the responses of the multiple participants and potentially allow for more than two contributors to the conversation. As of now the best metric available is human evaluation, this was not used in this paper, but it would shed more light on the competency of chatbot dialogue and would be a very useful metric in further work. All of these possible future topics could bring human level computer dialogue generation closer to reality.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1659788. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References


Table 2: Samples of generated text using the NET model showing improved coherence and cohesion with a larger training corpus (10,000) as compared with a very small corpus (100).

Trained on 100 sentences from SCOTUS
"JMTM SLIE : unt the the aoe the co satii the as io aoe tooos con is tuue te te cat ot tian toitoe on te tete an ao the ean to ar anert ao the the so tnt cot aorey sae thot ao the to cait ar te teat oa that ens ae ao thos wo toe ton fe the contits te see tun tee aoe tont cot sh the ian se tee an that tonttis the uant aae tin an th a oita enntte io on tae se the oan toe tot ton as por aor iie cot ao thl bid the san thee the oo to the tce fortte ior an aar sos aoint te talt oh uortee innt aoit cors ite o""


Templin, M. C. 1957. Certain language skills in children; their development and interrelationships.


