ANALYSING METHODS OF NEURAL TEXT GENERATION TO REFINE CONVERSATIONS

Mihir Antwal
19BCE1641
Vellore Institute of Technology.
Chennai, India
mihir.antwal2019@vitstudent.ac.in

Sam Methuselah
19BCE1698
Vellore Institute of Technology.
Chennai, India
sam.methuselah2019@vitstudent.ac.in

Abstract

A good conversation requires a combination of simplicity and detail, staying on topic and shifting it, and asking and answering questions. Although human evaluations of overall quality are routinely used to evaluate dialogue agents, the relationship between overall quality and these individual factors are not well-studied. In this paper, we look at conditional training and weighted decoding as two adjustable neural text generation strategies for controlling four dialogue attributes: repetition, specificity, response-relatedness, and question-asking. We use the PersonaChat task to undertake a large-scale human evaluation to see how these control factors affect multi-turn interactive talks. We examine their relationship to high-level elements of discourse in depth, demonstrating that manipulating combinations of these variables improves human quality assessments significantly.

I. Introduction

Despite their prevalence in contemporary research, neural generation models for discourse remain poorly understood. Many well-known issues, such as the genericity and repetition of responses (Serban et al., 2016a), are still unsolved. Surprisingly, little research has been done on the aspects that influence human assessments of overall conversation quality. Most studies have focused on the difficulty of predicting the next utterance, whereas evaluating the quality of a full discussion requires a multi-turn evaluation.

We conduct a large-scale investigation to determine the fine-grained parameters driving human judgements of full conversations, and we construct models that apply our findings in practise, resulting in state-of-the-art performance. We define and investigate eight features of conversation that may be quantified by human judgements, as well as four types of low-level attributes that can be algorithmically controlled in neural models. We explore two simple but general strategies for controlling low-level model attributes: conditional training, in which the neural model is conditioned on extra control features, and weighted decoding, in which control variables are only introduced to the decoding score function at test time.

The relevance of conversational flow has been overlooked in prior research, as traditional models repeat or contradict previous remarks, fail to balance specificity with genericness, and fail to balance asking questions with other dialogue activities. We achieve significantly greater

engagingness scores than the baseline by optimising control of repetition, specificity, and question-asking throughout numerous turns while conducting trials on the PersonaChat task (Zhang et al.,2018b).

II. Related Work

Automatic measures and their relationship to human judgments are less well understood in chitchat tasks, which we investigate in this paper. Because of the open-ended nature of dialogue, word-overlap metrics are successful for question-answering and machine translation, but they have little to no correlation with human assessments (Liu et al., 2016; Novikova et al., 2017). Recent attempts to improve automatic techniques, such as adversarial evaluation (Li et al., 2017b) and learning a scoring model (Lowe et al., 2017), have yielded mixed results.

Nonetheless, a handful of studies rely solely on automated measures, with no human research involved (Lowe et al., 2015; Parthasarathi and Pineau, 2018; Serban et al.,2016b). Other studies (Dinan et al., 2018; Li et al., 2016a,b; Venkatesh et al., 2017; Vinyals and Le, 2015; Zhang et al., 2018b) do utilise human evaluations, but they often report only one type of judgement (quality or appropriateness) via a Likert scale or pairwise comparison. Rather than comprehensive multi-turn dialogue, most of those efforts merely consider single turn evaluations, frequently with a reduced dialogue history.

In goal-oriented tasks, where automated procedures can be coded by evaluating task completion, dialogue assessment is relatively well understood (Bordes et al., 2017; El Asri et al., 2017; Hastie, 2012; Henderson et al., 2014; Wenet al., 2017). The PARADISE architecture can link task success and dialogue cost to human judgements like user pleasure (Walker et al., 1997).

By combining multiple automatic metrics designed to capture various conversational aspects (engagement, coherence, domain coverage, conversational depth, and topical diversity) within the scope of the Alexa prize (Venkatesh et al., 2017; Guo et al., 2018), a more comprehensive evaluation strategy has been studied (Venkatesh et al., 2017; Guo et al., 2018). We also focus on lower-level features (e.g. avoiding repetition, fluency) to understand how they correspond to both our controllable traits and overall quality assessments, despite the fact that they are comparable to the aspects investigated here.

Researchers have proposed a number of methods for controlling sentiment, length, speaker style, and tense in RNN-based natural language creation (Fan et al., 2018; Ficler and Goldberg, 2017; Ghazvininejad et al., 2017; Hu et al., 2017; Kikuchi et al., 2016; Peng et al., 2018; Wang et al., 2017). In the context of single-turn response generation, various works use control to address the same common sequence-to-sequence concerns we address here (especially genericness and unrelated output) (Baheti et al., 2018; Li et al., 2016a, 2017a; Shen et al., 2017; Xing et al., 2017; Zhang et al., 2018a; Zhou et al., 2017). We, on the other hand, concentrate on developing controls for multi-turn interactive dialogue and human evaluation of it, which includes a new mechanism for controlling attributes at the dialogue level rather than the utterance level.

We need a control approach that is both general-purpose (one technique to control multiple qualities at the same time) and easily customizable for this project (the control setting is adjustable after training). Given these limits, we investigate two control methods: conditional

training (Fan et al. (2018); Kikuchi et al. (2016); Peng et al. (2018)) and weighted decoding (presented by Ghazvininejad et al. (2017) as a generic methodology and by Baheti et al. (2018) to control response-relatedness). This is the first study to comprehensively assess the effectiveness of two general-purpose control approaches across numerous attributes, to our knowledge.

III. Methodology

1. Baseline model

A 2-layer LSTM sequence-to-sequence model with attention serves as our baseline model. The whole discussion history (separated using distinct speaker-identifying tokens) is input x to the encoder on any dialogue turn, with the model's own persona prepended. The decoder generates a response y based on the input sequence x. Unless otherwise noted, all of our models decode utilising beam search with a beam size of 20.

We used 300-dimensional GloVe embeddings to start the word embedding matrix (Pennington et al., 2014). We pre-trained the model using a dataset of 2.5 million Twitter message-response pairings using the ParlAI framework (Miller et al., 2017), then fine-tuned it on PersonaChat. The baseline model has a perplexity of 26.83 and an F1 of 17.02 on the PersonaChat validation set, which would have placed us 4th out of 26 models in the ConvAI2 competition (Dinan et al., 2019). Using control, we try to improve on this baseline.

2. Controllable text generation methods

Assume we have a sequence-to-sequence model with P(y|x) = tP(yt |x, y1,..., yt1), which is the conditional probability of a response y (the model's next utterance) given input x (the context, which in our instance comprises the model's own persona and the dialogue history).

Unlike earlier work, which controls attributes of the output y at the sentence level, we want to control attributes of the output y at the dialogue level — that is, we want to utilise a single control setting for an entire dialogue. For example, rather than providing a control setting for each utterance (e.g. is a question or isn't a question), we provide a control setting at the beginning of each discussion (e.g. 20 percent questions or 70 percent questions). The sequence-to-sequence model can pick what value the controlled attribute should take for any given utterance, but we can choose the overall distribution with this approach. The sequence-to-sequence model, for example, is often good at determining when to ask a question, therefore we find that this strategy works well. This is especially easier than the alternative, which is to create a distinct mechanism for deciding whether to ask a question for each statement.

In this section, we'll go through the two approaches we utilise to influence the properties of the output y at the conversation level, which we name Conditional Training (CT) and Weighted Decoding (WD).

a. Conditional Training (CT)

Conditional Training is a method for learning a sequence-to-sequence model P(y|x, z), where z is a discrete control variable (Fan et al., 2018; Kikuchi et al., 2016; Peng et al., 2018). We utilise z to express bucketed ranges where the control attribute is naturally continuous (for example, repetitiveness, specificity, and response-relatedness in our work). z represents an overall probability for a binary property like question-asking.

To train a CT model, we annotate every (x, y) pair in the training set with the characteristic we want to regulate automatically (for example, whether y contains a question mark). During training, we calculate the corresponding z value for each sample. After that, an embedding is used to represent the control variable z. (each of the possible values of z has its own embedding). The embedding length in all of our trials is 10; this was found through hyperparameter optimization. There are a few ways to condition the sequence-to-sequence model on z, such as appending z to the end of the input sequence or using z as the decoder's START symbol. On every stage, we find it most advantageous to concatenate z to the decoder's input. 2 Finally, the CT model optimises the cross-entropy loss to obtain y = y1,..., yT:

$$\text{LossCT} = -\frac{1}{T}\sum_{t=1}^{T} \log \mathrm{P}(y_t \,|\mathrm{x}, \mathrm{z}, y_1, \ldots, y_{t-1})$$

Our CT models are fine-tuned to optimise lossCT on the PersonaChat training set, until lossCT convergence on the validation set, using the parameters from the baseline sequence-to-sequence model P(y|x) (the new decoder parameters are initialised with small random values).

b. Weighted Decoding (WD)

Weighted Decoding (Ghazvininejad et al., 2017) is a decoding technique that enhances or decreases the likelihood of words with specific characteristics. The methodology is solely used during testing and does not require any changes to the training procedure. The controlled attribute must be defined at the word level in WD, hence any desired utterance-level attribute must be redefined via word-level features. In weighted decoding, a partial hypothesis yt = y1,..., yt1 is expanded on the t^{th} step of decoding by computing the score for each possible next word w in the vocabulary:

$$\text{score}(w , y{<}t ; x) = \text{score}(y{<}t ; x) + \log \text{PRNN}(w|y{<}t , x) + \sum_i \; wi * fi(w ; y{<}t , x)$$

The log-probability of the word w determined by the RNN is log PRNN (w|yt, x), the accumulated score of the already-generated words in the hypothesis yt is score(yt; x), and the decoding features with associated weights wi are fi(w; yt, x). Numerous features fi (to control multiple attributes) are possible, and the weights wi are hyperparameters that must be selected.

In the context of the text generated so far yt and the context x, a decoding feature fi(w; yt, x) assigns a real value to the word w. The characteristic can be continuous (for example, whether w starts with the same letter as the final word in yt), discrete (for example, the length of w in characters), or binary (for example, whether w starts with the same letter as the last word in yt). A positive weight wi raises the probability of words w scoring highly in relation to fi, while a negative weight lowers it.

It's worth noting that weighted decoding and conditional training can both be used at the same time (for example, train a CT model and then apply WD at test time), which is an approach we employ in our research.

IV. Controlling conversational attributes

We detail how conditional training and weighted decoding are used to manage four attributes: repetition, specificity, response-relatedness, and question-asking in this section. We employ automatic metrics to assess the success of both control methods (i.e., how well the attribute was controlled), and our findings are used to identify control methods and control settings to be investigated further via human review.

1. Repetition

External repetition (self-repetition across utterances), internal repetition (self-repetition within utterances), and partner repetition are the three types of repetition we see in our baseline model (repeating the conversational partner). We define five n-gram based decoding features to control repetition with weighted decoding. For the three repetition kinds, three of these traits (extrep_bigram, intrep_bigram, and partnerrep_bigram) identify repeated bigrams. The remaining two features (extrep_unigram and intrep_unigram) identify content words that repeat. We can reduce repetition by giving these qualities a negative weight. In particular, if the weight is −∞, our method is equivalent to Kulikov et aln-gram .'s blocking (2018). We've seen that repetition control is crucial, thus we've included it in all subsequent control studies.

2. Specificity

Our baseline, like many other sequence-to-sequence models that use beam search decoding, commonly asks basic queries like What music do you prefer? I like various sorts of music, for example, and delivers bland, unspecific comments.

As a measure of word rarity, we use the Normalized Inverse Document Frequency (NIDF) to manage specificity.

IDF(w) = log(R/cw), where R is the number of replies in the dataset and cw is the number of those responses that contain w, is the Inverse Document Frequency of a word w. Normalized IDF (which ranges from 0 to 1) is

$$NIDF(w) = (IDF(w) - min\_idf) / (max\_idf - min\_idf)$$

where min idf and max idf are the least and maximum IDFs, respectively, over the entire vocabulary. We employ NIDF as a decoding feature to manage specificity with weighted decoding. This approach generates reasonable outputs when the feature weight is within a particular range, as indicated in Table 1, but nonsensical outputs at the extremes. The limit for nonsensical output varies from one example to the next.

Input: *Yes, I'm studying law at the moment*
Baseline Response: *That sounds like a lot of fun!*

Wt	NIDF	Weighted Decoding Response
-5.0	0.6%	*Oh.....*
0.0	17.1%	*That sounds like a lot of fun!*
3.0	18.3%	*That sounds like a lot of fun. How long have you been studying?*
7.0	38.5%	*I majored in practising my spiritual full time philosophy test*
10.0	71.9%	*Oh wow! Merna jean isa paino yi hao hui bu acara sya gila [. . .]*
z	**NIDF**	**Conditional Training Response**
0	16.8%	*Sounds like you are a great person!*
2	18.3%	*So you are a law student?*
4	18.4%	*That sounds like a lot of fun*
6	22.8%	*That sounds like a rewarding job!*
8	24.4%	*That sounds like a rewarding career!*

Table 1: Middle: Example of controlling specificity (NIDF) via weighted decoding. At the extremes, the model produces only the rarest or the most common tokens. Bottom: Example of controlling specificity via conditional training. This gives a narrower NIDF range, but all the responses are appropriate.

We define the specificity of an utterance y as the mean NIDF of the words in y to control specificity using conditional training. As a result, our control variable z is the mean NIDF (discretized into 10 equal-sized buckets). This method yields outputs with a narrower NIDF range, but less nonsensical outcomes overall, as seen in Table 1.

3. Response-relatedness

In discussion, it's normally preferable to respond with something connected to the partner's most recent statement; for example, if the partner says, "My grandfather died last month," it's customary to respond, "I'm so sorry." Did you have a close relationship with your grandfather? Our baseline model, on the other hand, frequently responds with unrelated phrases such as Do you have any pets?

We use the decoding feature *resp_rel:* to control response-relatedness when using weighted decoding.

$$\text{resp\_rel}(w;\ y_{<t},\ x) = \text{cos\_sim}(\text{word\_emb}(\omega), \text{sent\_emb}(Y))$$

where word emb(w) is the GloVe embedding for the word w, sent emb(Y) is the sentence embedding for the partner's most recent utterance Y (note that Y is part of the context x), and cos sim is the cosine similarity between them. The sentence embedding sent emb(s) for an utterance s is, in particular, a weighted average of the GloVe embeddings of the words in s, with the first principal component projected out;

see Arora et al. for more information (2017). This method of response-relatedness management is similar to the one described in (Baheti et al., 2018). We discovered that using weighted decoding to adjust the semantic relatedness of the model's reaction to the partner's most recent utterance is effective (see Table 2). We discover that excessive weights produce nonsensical results, as we did before.

We try defining the control variable z as cos_sim(sent_emb(y), sent_emb(Y')), the overall cosine similarity between the partner's previous utterance ' and the model's response y, to manage response-relatedness with conditional training (again, we discretize z). However, we believe that this strategy is useless, as the CT model only learns a very weak link between z and the output's semantic relatedness

Input: *Do you go get coffee often*
Baseline Response: *I do, when I am not playing the piano.*

Wt	Sim	Weighted Decoding Response
-10.0	-0.05	*I am a musician.*
0.0	-0.02	*I do, when I am not playing the piano.*
5.0	0.40	*I do, usually at starbucks.*
8.0	0.59	*Not usually, especially when you drink latte.*
11.0	0.72	*Not often, usually with drinks, espresso, latte, tea, etc.*

Table 2: Example of controlling response-relatedness (cosine similarity to input) via weighted decoding. Positive weights (e.g. 5.0) can yield more on-topic responses, but higher weights (e.g. 11.0) can result in nonsensical lists of topically related words.

4. Question-asking

A reciprocal asking and answering of questions is required for considerate chitchat; asking too few or too many questions can sound self-centred or inquisitive. To explore these trade-offs, we limit the number of questions we ask.

We employ the binary decoding feature is qn word(w), which is equal to 1 if and only if the word w is in a pre-defined list of interrogative terms (how, what, when, where, which, who, whom, whose, why?) to regulate question-asking using weighted decoding. We've found that a negative weight can inhibit genuine non-question utterances that happen to contain interrogative phrases (such as I'm learning to knit), while a positive weight can result in degenerate utterances (such as What? or Who? When? How?).

We consider an utterance y to have a question if and only if it contains a question mark for conditional training. Our CT model is trained on a control variable z that has 11 potential values: 0, 1,..., 10. We want to govern question-asking at the distributional, dialogue level, rather than the binary, utterance level. As a result, setting z = I means that the model should create utterances containing '?' with a probability of i/10 on average. We allocate instances to buckets at random during training so that each bucket I gets trained on examples with the right proportion of questions (i/10), and all buckets have the same number of training examples.

Conditional training is found to be effective in controlling question-asking — as shown in Figure 2, raising z from 0 to 10 results in a range of question-asking rates ranging from 1.40 percent to 97.72 percent. Question-asking is lowered when repetition control is introduced; for example, the z = 10 setting (which should give 100% questions) now only produces 79.67% questions. The main issue is extrep_bigram, a weighted decoding feature that inhibits bigrams that have appeared in prior utterances, preventing the model from producing bigrams that are prevalent in many inquiries, such as do you and what is. To address this, we add a new setting z = 10 (boost), in which we don't utilise the feature extrep_bigram for weighted decoding during beam search but do use it to rerank the candidates after beam search. This configuration, which allows the model to generate the required question-asking bigrams, results in a 99.54 percent question-asking rate at the cost of somewhat increased external bi gram repetition.

Conditional training is better than weighted decoding for managing question-asking for two reasons. For starters, it enables us to achieve (near) 0 percent questions, 100% questions, or anything in between without risking degenerate output. Second, the presence of a question mark more precisely and immediately represents the true attribute of interest (question-asking) than the inclusion of interrogative terms. As a result, only the CT approach is taken into account in the human evaluation.

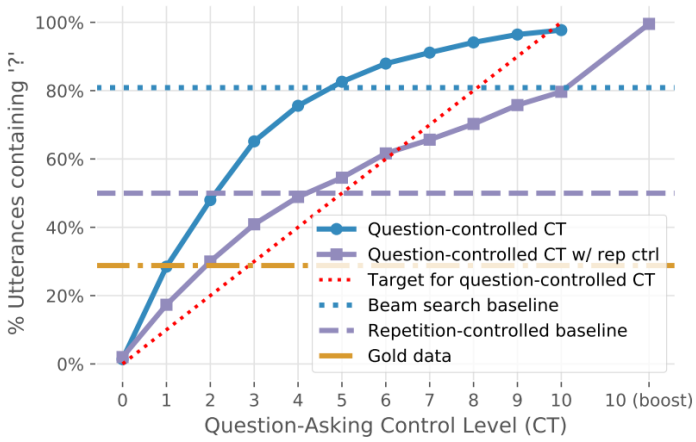

Figure 2: Controlling question-asking via conditional training. Exact numbers can be found in Appendix F.

V. Comparison of control methods

Conditional training and weighted decoding are both valuable strategies, as seen in the previous section, but they have different benefits and disadvantages.

The most significant disadvantage of conditional training is that it occasionally fails to learn the relationship between the control variable z and the desired output y. In practise, we've discovered that the model can learn simple output features (such as the existence of '?' and overall genericness), but not input-output correlations (such as semantic relatedness). Weighted decoding, on the other hand, can force the desired feature to appear in the output by arbitrarily increasing the weight (though this may have unintended side-effects).

The main drawback of weighted decoding is that it runs the danger of going off-distribution if the weight is too high. Conditional training, on the other hand, yields generally well-formed, in-distribution outputs. This emphasises the importance of learnt control: it is better to learn to

create output that satisfies the control variable while also being suitable, rather than changing the decoding process to fulfil the control variable while potentially sacrificing appropriateness in the process.

Other considerations include:

(1) Convenience: conditional training requires retraining, whereas weighted decoding does not, but is slower at test time.
(2) Data accessibility: conditional training necessitates the use of training instances of the controlled characteristic, whereas weighted decoding can control any computed feature without the use of examples.
(3) Definition of attributes: Conditional training can control attributes at the sentence level, but they must be discrete. Weighted decoding, on the other hand, necessitates word-level characteristics, which can be continuous.

VI. Human evaluation results

We perform a large-scale human evaluation of 28 model configurations, as well as human-human dialogues for comparison, in order to evaluate the effect of our configurable qualities.

Controlling for repetition, specificity, and question-asking all result in significant engagement improvements over the greedy and beam-search baseline models, as seen in Figure 3. Controlling for multi-turn (self) repetition is particularly critical, and it should be included alongside other attribute control approaches. Controlling response-relatedness resulted in no improvement.

We evaluate the whole collection of human assessments, as shown in Figure 4, to further understand these overall engagingness gains. We've discovered that decreasing repetition improves all areas of conversational quality. Over the repetition-controlled baseline, increasing specificity improves interestingness and listening ability, while increasing question-asking improves inquisitiveness and curiosity.

Our most engaging model, which controls both repetition and question-asking and is labelled 'Question (CT)' in Figure 3 (left), is on par with the winning entry in the ConvAI2 competition, with a raw score7 of 3.1. (Dinan et al., 2019). The ConvAI2 winner, Lost in Conversation, was trained on almost 12 times the amount of data as our model. The OpenAI GPT Language Model (Radford et al., 2018) is based on the BookCorpus (Zhu et al., 2015), which comprises roughly 985 million words, whereas our model is built on the Twitter dataset (Radford et al., 2018). (approximately 79 million words).

Overall, our findings demonstrate that managing low-level qualities over numerous turns improves overall quality.

Model	Win%	Top 3 reasons for preferring model
Specificity WD (weight = 6)	84.1%	*More information; Better flow; More descriptive*
Specificity WD (weight = 4)	75.5%	*More information; They describe their life in more detail; Funny*
Specificity CT ($z = 7$)	56.2%	*More information; Better flow; Seems more interested*

Table 3: A/B tests comparing various specificity-controlled models to the repetition-controlled baseline on interestingness. We find all comparisons are significant ($p < .05$; binomial test).

We were surprised that our more particular models did not provide clearer improvements in interestingness, despite the fact that they yielded large improvements in engagingness. To find out more, we compared three specificity-controlled models to the repetition-controlled baseline in an A/B interestingness test. Crowdworkers were shown two discussions (from the primary human evaluation) and asked to pick the more fascinating model (see Figure 7 for details). We gathered 500 samples for each comparison, as well as 200 additional human vs. repetition-controlled baseline samples for quality control. We have about 300 evaluations per comparison after eliminating low-quality crowdworkers, with an average Cohen's 0.6.

All three models were deemed much more intriguing than the repetition-controlled baseline, as seen in Table 3. This persuasively demonstrates that using more uncommon words in utterances is a viable method for increasing interest. We have two theories as to why these disparities in interestingness did not show up in our main evaluation. For starters, unlike more practical measurements like avoiding repetition and making logic, interestingness is a highly subjective statistic, making it difficult to calibrate across crowdworkers.

Second, we believe that the crowdworkers may have rated the task's interestingness rather than the chatbot in our original evaluation. This could explain why minor improvements in conversational abilities did not result in better interestingness ratings – the PersonaChat assignment has a built-in restriction on how fascinating it can be.

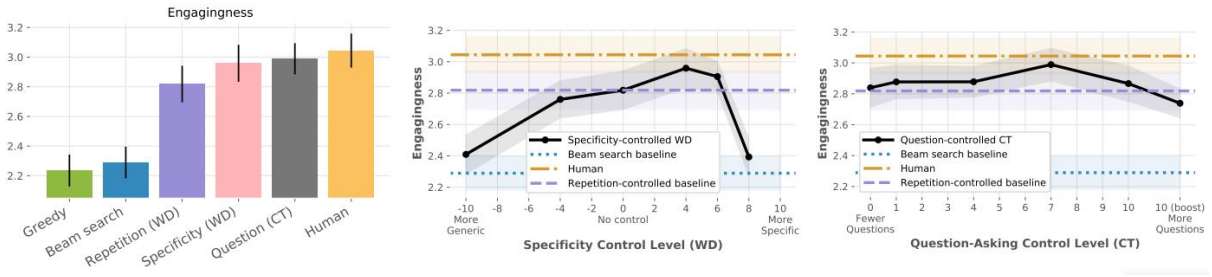

Figure 3: Calibrated human judgments of engagingness for the baselines and best controlled models (left); for different specificity control settings (middle); and for different question-asking control settings (right).

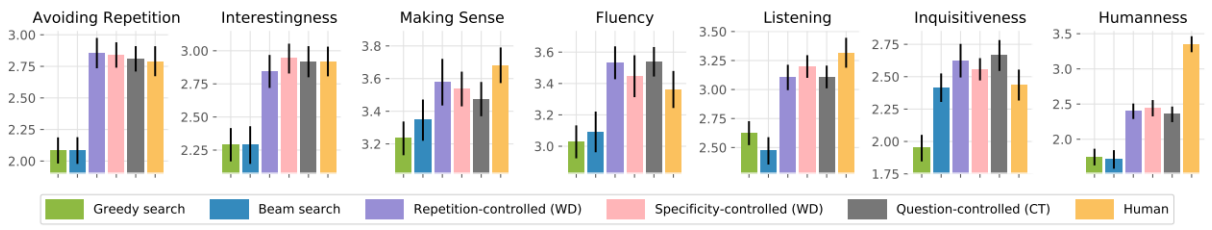

Figure 4: Calibrated human judgments of conversational aspects for the baselines and best controlled models. Note: In Figure 3 and here, the Specificity and Question controlled models both include Repetition control, but Question control doesn't include Specificity control, or vice versa.

VII. Conclusion

We discovered that a good discussion is all about balance — ensuring the correct amount of repetition, specificity, and question-asking is crucial for overall quality. We also discovered that conversational characteristics like interest, listening, and inquisitiveness are crucial — albeit improving these can result in a trade-off with certain types of errors (such as repetitive, disfluent, or nonsensical output). Second, multiturn evaluation is required to determine what makes a good discussion — numerous turns are required to uncover flaws like repetition, consistency, and frequency of question-asking. Finally, what do we mean when we say "good"? Although both humanness and engagingness are widely employed as general quality indicators, they are not the same. While our models scored near-human on engagingness, they fell short on humanness, demonstrating that a chatbot does not have to be human-like to be fun.

Our research reveals that when applied to open-ended discussion, neural generating systems have systemic issues, some of which (for example, repetition) are only visible in the multi-turn situation. Furthermore, controlling low-level attributes is a viable strategy to address these issues, resulting in significant increases in overall quality — equivalent to systems trained on far more data in our situation. Future work will include automatically optimising control settings and creating more convincingly human-like chatbots.

References

[1] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In Proceedings of the International Conference on Learning Representations (ICLR).

[2] Ashutosh Baheti, Alan Ritter, Jiwei Li, and Bill Dolan. 2018. Generating more interesting responses in neural conversation models with distributional constraints. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3970–3980. Association for Computational Linguistics.

[3] Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In Proceedings of the International Conference on Learning Representations (ICLR).

[5] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of Wikipedia: Knowledgepowered conversational agents. arXiv preprint arXiv:1811.01241.

[6] Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: a corpus for adding memory to goal-oriented dialogue systems. In Proceedings of the 18th Annual SIGDIAL Meeting on Discourse and Dialogue, pages 207–219, Saarbrucken, Germany. Association for Com- ¨ putational Linguistics.

[7] Angela Fan, David Grangier, and Michael Auli. 2018. Controllable abstractive summarization. In Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, pages 45–54. Association for Computational Linguistics.

[8] Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural languagegeneration. In Proceedings of the Workshop on Stylistic Variatio n, pages 94–104. Associationfor Computational Linguistics.

[9] Marjan Ghazvininejad, Xing Shi, Jay Priyadarshi, and Kevin Knight. 2017. Hafez: an interactive poetry generation system. In Proceedings of ACL 2017, System Demonstrations, pages 43– 48. Association for Computational Linguistics.

[10] Fenfei Guo, Angeliki Metallinou, Chandra Khatri, Anirudh Raju, Anu Venkatesh, and Ashwin Ram. 2018. Topic-based evaluation for conversational bots. Advances in Neural Information Processing Systems, Conversational AI Workshop.

[11] Helen Hastie. 2012. Metrics and evaluation of spoken dialogue systems, pages 131–150. Springer.

[12] Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The second dialog state tracking challenge. In Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), pages 263–272.

[13] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In Thirty-fourth International Conference on Machine Learning.

[14] Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoderdecoders. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 328–1338. Association for Computational Linguistics.

[15] Ilya Kulikov, Alexander H Miller, Kyunghyun Cho, and Jason Weston. 2018. Importance of a search strategy in neural dialogue modelling. arXiv preprint arXiv:1811.00907.

[16] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversitypromoting objective function for neural conversation models. In Proceedings of the

2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 110–119. Association for Computational Linguistics.

[17] Jiwei Li, Will Monroe, and Dan Jurafsky. 2017a. Learning to decode for future success. arXiv preprint arXiv:1701.06549.

[18] Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016b. Deep 1712 reinforcement learning for dialogue generation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1192–1202, Austin, Texas. Association for Computational Linguistics.

[19] Jiwei Li, Will Monroe, Tianlin Shi, Sebastien ´Jean, Alan Ritter, and Dan Jurafsky. 2017b. Adversarial learning for neural dialogue generation. arXiv preprint arXiv:1701.06547.

[20] Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. pages 2122–2132.

[21] Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic turing test: Learning to evaluate dialogue responses. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1116–1126. Association for Computational Linguistics.

[22] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The Ubuntu dialogue corpus: A large dataset for research in unstructured multiturn dialogue systems. In Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 285– 294, Prague, Czech Republic. Association for Computational Linguistics.

[23] Alexander Miller, Will Feng, Dhruv Batra, Antoine Bordes, Adam Fisch, Jiasen Lu, Devi Parikh, and Jason Weston. 2017. ParlAI: A dialog research software platform. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 79–84, Copenhagen, Denmark. Association for Computational Linguistics.

[24] Jekaterina Novikova, Ondˇrej Dusek, Amanda Cer- ˇcas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for nlg. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2241–2252.

[25] Prasanna Parthasarathi and Joelle Pineau. 2018. Extending neural generative conversational model using external knowledge sources. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 690–695, Brussels, Belgium. Association for Computational Linguistics.

[26] Nanyun Peng, Marjan Ghazvininejad, Jonathan May, and Kevin Knight. 2018. Towards controllable story generation. In Proceedings of the First Workshop on Storytelling, pages 43–49. Association for Computational Linguistics.

[27] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532– 1543, Doha, Qatar. Association for Computational Linguistics.

[28] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

[29] Iulian Vlad Serban, Ryan Lowe, Laurent Charlin, and Joelle Pineau. 2016a. Generative deep neural networks for dialogue: A short review. Advances in Neural Information Processing Systems workshop on Learning Methods for Dialogue.

[30] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. 2016b. Building end-to-end dialogue systems using generative hierarchical neural network models. In AAAI, volume 16, pages 3776–3784.

[31] Xiaoyu Shen, Hui Su, Yanran Li, Wenjie Li, Shuzi Niu, Yang Zhao, Akiko Aizawa, and Guoping Long. 2017. A conditional variational framework for dialog generation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 504–509. Association for Computational Linguistics.

[32] Anu Venkatesh, Chandra Khatri, Ashwin Ram, Fenfei Guo, Raefer Gabriel, Ashish Nagar, Rohit Prasad, Ming Cheng, Behnam Hedayatnia, Angeliki Metallinou, et al. 2017. On evaluating and comparing conversational agents. Advances in Neural Information Processing Systems, Conversational AI Workshop.

[33] Oriol Vinyals and Quoc Le. 2015. A neural conversational model. In Proceedings of the 31st International Conference on Machine Learning, Deep Learning Workshop, Lille, France.1713

[34] Marilyn A. Walker, Diane J. Litman, Candace A. Kamm, and Alicia Abella. 1997. PARADISE: A framework for evaluating spoken dialogue agents. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, pages 271–280, Madrid, Spain. Association for Computational Linguistics.

[35] Di Wang, Nebojsa Jojic, Chris Brockett, and Eric Nyberg. 2017. Steering output style and topic in neural response generation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2140– 2150. Association for Computational Linguistics.

[36] Tsung-Hsien Wen, David Vandyke, Nikola Mrksi˘ c, Milica Gasic, Lina M. Rojas Barahona, ´ Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 438–449. Association for Computational Linguistics.

[37] Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In AAAI, volume 17, pages 3351–3357.

[38] Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, Jun Xu, and Xueqi Cheng. 2018a. Learning to control the specificity in neural response generation. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1108–1117, Melbourne, Australia. Association for Computational Linguistics.

[39] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston.2018b. Personalizing dialogue agents: I have a dog, do you have pets too? In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

[40] Ganbin Zhou, Ping Luo, Rongyu Cao, Fen Lin, Bo Chen, and Qing He. 2017. Mechanismaware neural machine for dialogue response generation. In AAAI, pages 3400–3407.

[41] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In Proceedings of the IEEE international conference on computer vision, pages 19–27.