Improving Neural Conversational Models with Entropy-Based Data Filtering

Richárd Csáky
Department of Automation
and Applied Informatics
Budapest University of
Technology and Economics
ricsinaruto@hotmail.com

Patrik Purgai
Department of Automation
and Applied Informatics
Budapest University of
Technology and Economics
purgai.patrik@gmail.com

Gábor Recski
Apollo.ai
gabor@apollo.ai

Abstract

Current neural-network based conversational models lack diversity and generate boring responses to open-ended utterances. *Priors* such as persona, emotion, or topic provide additional information to dialog models to aid response generation, but annotating a dataset with priors is expensive and such annotations are rarely available. While previous methods for improving the quality of open-domain response generation focused on either the underlying model or the training objective, we present a method of filtering dialog datasets by removing generic utterances from training data using a simple entropy-based approach that does not require human supervision. We conduct extensive experiments with different variations of our method, and compare dialog models across 13 evaluation metrics to show that training on datasets filtered in this way results in better conversational quality as chatbots learn to output more diverse responses.

1 Introduction

Current open-domain neural conversational models (NCM) are trained on pairs of source and target utterances in an effort to maximize the likelihood of each target given the source (Vinyals and Le, 2015). However, real-world conversations are much more complex, and a plethora of suitable targets (responses) can be adequate for a given input. We propose a data filtering approach where the "most open-ended" inputs - determined by calculating the entropy of the distribution over target utterances - are excluded from the training set. We show that dialog models can be improved using this simple unsupervised method which can be applied to any conversational dataset. In addition, we conduct several experiments to uncover how some of the current open-domain dialog evaluation methods behave with respect to overfitting and random data. We plan to release all experimental code and data. Filtering other datasets is also straightforward with our code.

2 Background

Open-domain NCMs are mostly based on neural network architectures developed for machine translation (MT, Sutskever et al. (2014); Cho et al. (2014); Vaswani et al. (2017)). Conversational data differs from MT data in that targets to the same source may vary not only grammatically but also semantically (Wei et al., 2017; Tandon et al., 2017): consider plausible replies to the question "What did you do today?". Dialog datasets also contain generic responses, e.g. "yes", "no" and "i don't know", that appear in a large and diverse set of contexts (Mou et al., 2016; Wu et al., 2018). Following the approach of modeling conversation as a sequence to sequence (`seq2seq`, Sutskever et al. (2014)) transduction of single dialog turns, these issues can be referred to as the *one-to-many*, and *many-to-one* problem. `seq2seq` architectures are not suited to deal with the ambiguous nature of dialogs since they are inherently deterministic, meaning that once trained they cannot output different sequences to the same input. Consequently they tend to produce boring and generic responses (Li et al., 2016a; Wei et al., 2017; Shao et al., 2017; Zhang et al., 2018a; Wu et al., 2018).

Previous approaches to the *one-to-many*, *many-to-one* problem can be grouped into three categories. One approach involves feeding extra information to the dialog model such as dialog history (Serban et al., 2016; Xing et al., 2018), categorical information like persona (Li et al., 2016b; Joshi et al., 2017; Zhang et al., 2018b), mood/emotion (Zhou et al., 2018; Li et al., 2017b), and topic (Xing et al., 2017; Liu et al., 2017; Baheti et al., 2018), or through knowledge-bases (Dinan et al., 2019; Ghazvininejad et al., 2018; Zhu et al., 2017). A downside to these approaches is that they require annotated datasets which are not always available, or might be smaller in size. Augmenting the model itself, with e.g. latent variable sampling (Serban et al., 2017b; Zhao et al., 2017, 2018; Gu et al., 2019; Park et al., 2018; Shen et al., 2018b), or improving the decoding process (Shao et al., 2017; Kulikov et al., 2018; Mo et al., 2017; Wang et al., 2018) is also a popular approach. Sampling provides a way to generate more diverse responses, however such models are more likely to output ungrammatical or irrelevant responses. Finally, directly modifying the loss function (Li et al., 2016a), or training by reinforcement (Li et al., 2016d; Serban et al., 2017a; Li et al., 2016c; Lipton et al., 2018; Lewis et al., 2017) or adversarial learning (Li et al., 2017a; Ludwig, 2017; Olabiyi et al., 2018; Zhang et al., 2018c) has also been proposed, but this is still an open research problem, as it is far from trivial to construct objective functions that capture conversational goals better than cross-entropy loss.

Improving dataset quality through filtering is frequently used in the machine learning literature (Sedoc et al., 2018; Ghazvininejad et al., 2018; Wojciechowski and Zakrzewicz, 2002). Xu et al. (2018b) introduced coherence for measuring the similarity between contexts and responses, and then filtered out pairs with low coherence. This improves datasets from a different aspect and could be combined with our present approach. However, natural conversations allow many adequate responses that are not similar to the context, thus it is not intuitively clear why filtering these should improve dialog models. Our experiments also further support that cross-entropy is not an adequate loss function (shown qualitatively by Csáky (2017) and Tandon et al. (2017)), by showing that many automatic metrics continue to improve after the validation loss reaches its minimum and starts increasing. However, we found that the metrics steadily improve even after we can be certain that the model overfitted (not just according to the loss function). Further research is required, to determine whether this indicates that overfitted model responses are truly better or if it's a shortcoming of the metrics that they prefer such models (see Section 5 for more discussion).

Currently, there is no well-defined automatic evaluation method (Liu et al., 2016), and while some metrics that correlate more with human judgment have been proposed recently (Li et al., 2017a; Lowe et al., 2017; Tao et al., 2018), they are harder to measure than simpler automatic metrics like perplexity or BLEU (Papineni et al., 2002). Furthermore, even human evaluation has its downsides, like high variance, high cost, and difficulty of replicating experimental setups (Zhang et al., 2018b; Tao et al., 2018). Some researches resort to only human evaluations (Krause et al., 2017; Fang et al., 2018), others use only automatic metrics (Olabiyi et al., 2018; Xing and Fernández, 2018; Kandasamy et al., 2017; Shalyminov et al., 2018; Xu et al., 2018b), and still others use both (Shen et al., 2018a; Xu et al., 2018a; Baheti et al., 2018; Ram et al., 2018). While we have not yet been able to conduct an extensive human evaluation (which is left for future work), we do conduct an especially thorough automatic evaluation both at the validation loss minimum and of overfitted models. We believe our experiments also shed light on the limitations of frequently used automatic metrics.

3 Methods

3.1 Intuition

We approach the *one-to-many*, *many-to-one* from a relatively new perspective: instead of adding more complexity to NCMs, we reduce the complexity of the dataset by filtering out a fraction of utterance pairs that we assume are primarily responsible for generic/uninteresting responses. Of the 72 000 unique source utterances in the DailyDialog dataset (see Section 4.1 for details), 60 000 occur with only a single target. For these it seems straightforward to maximize the conditional probability $P(T|S)$, S and T denoting a specific source and target utterance. However, in the case of sources that appear with multiple targets (*one-to-many*), models are forced to learn some "average" of observed responses (Wu et al., 2018).

The entropy of the distribution of responses to an utterance s is a natural measure of the amount of "confusion" introduced by s. For example the context "What did you do today?" has high entropy, since it is paired with many different responses in the data, but "What color is the sky?" has low entropy since it's observed with few responses. The *many-to-one* scenario can be similarly formulated, where a diverse set of source utterances are observed with the same target (e.g. "I don't know" has high entropy). While this may be a less prominent issue in training NCMs, we shall still experiment with excluding such generic targets, as dialog models tend to generate them frequently (see Section 2).

3.2 Clustering Methods and Filtering

We refer with IDENTITY to the following entropy computation method. For each source utterance s in the dataset we calculate the entropy of the distribution $T|S = s$, i.e. given a dataset D of source-target pairs, we define the *target entropy* of s as

$$H_{\text{tgt}}(s, D) = -\sum_{(s,t_i)\in D} p(t_i|s)\log_2 p(t_i|s) \quad (1)$$

Similarly, *source entropy* of a target utterance is

$$H_{\text{src}}(t, D) = -\sum_{(s_i,t)\in D} p(s_i|t)\log_2 p(s_i|t) \quad (2)$$

The probabilities are based on the observed relative frequency of utterance pairs in the data.

For the purposes of this entropy-based filtering we considered the possibility of also including some form of similarity measure between utterances that would allow us to detect whether a set of responses is truly diverse, as in the case of a question like "What did you do today?", or diverse only on the surface, such as in the case of a question like "How old are you?" (since numbers have similar embeddings). Furthermore, creating clusters and assigning entropy values to them can facilitate our goal of finding problematic utterances. For example "How are you?" can appear in many forms, like "How are you <name>?" (see Section 4.2). While separately the different forms have low entropy (because they have low frequency), we may decide to filter them all if together they form a high-entropy cluster.

To this end we performed the filtering based not only on the set of all utterances, as is the case with IDENTITY, but also on clusters of utterances established by clustering their vector representations using the Mean Shift algorithm (Fukunaga and Hostetler, 1975). Source and target utterances are clustered separately. In AVG-EMBEDDING the representation $R(U)$ of utterance U is computed by taking the average word embedding weighted by the smooth inverse frequency $R(U) = \frac{1}{|U|}\sum_{w\in U}\frac{E(w)\cdot 0.001}{0.001+p(w)}$ of words (Arora et al., 2017), where $E(w)$ and $p(w)$ are the embedding and the probability[1] of word w respectively. We also experiment with SENT2VEC[2], a more sophisticated sentence embedding approach, which can be thought of as an extension of word2vec to sentences (Pagliardini et al., 2018).

The *target entropy* of a source cluster c_s is

$$H_{\text{tgt}}(c_s, C) = -\sum_{(c_s,c_{t,i})\in C} p(c_{t,i}|c_s)\log_2 p(c_{t,i}|s) \quad (3)$$

where C is the set of all clusters and we sum over all target clusters $c_{t,i}$ in which respective targets are paired with the source utterances in c_s. Each utterance in a cluster is assigned the same entropy. Note that IDENTITY is a special case of this cluster-based entropy computation method, since in IDENTITY a "cluster" is comprised of multiple examples of one unique utterance. Thus a target cluster's entropy is computed similarly to Equation 2, but using clusters as in Equation 3.

For each of these methods for obtaining entropy values we experiment with 3 filtering strategies. The SOURCE approach filters utterance pairs in which the source utterance has high entropy, TARGET filters those with a high entropy target, and finally the BOTH strategy filters all utterance pairs that are filtered by either SOURCE or TARGET. Some additional techniques that did not yield meaningful improvement, and thus were excluded from further evaluation, include: 1. Clus-

[1] Probability is based on the observed relative frequency in the data.

[2] https://github.com/epfml/sent2vec

tering based on the Jaccard similarity of the bag of words of utterances. 2. Clustering only unique utterances. 3. Using K-means instead of Mean Shift. 4. Filtering stop words before clustering.

4 Data Analysis

4.1 Dataset

With 90 000 utterances in 13 000 dialogs, DailyDialog[3] (Li et al., 2017b), our dataset of choice, is comparable in size with the Cornell Movie-Dialogs Corpus, but contains high quality real-world dialogs instead of movie conversations, which are "not truthful representations of real-life conversations" (Danescu-Niculescu-Mizil and Lee, 2011). Using the IDENTITY approach, about 87% of utterances have 0 entropy (i.e. they do not appear with more than one target), 5% have an entropy of 1, remaining values rise sharply to about 7. This distribution is very similar for source and target utterances.

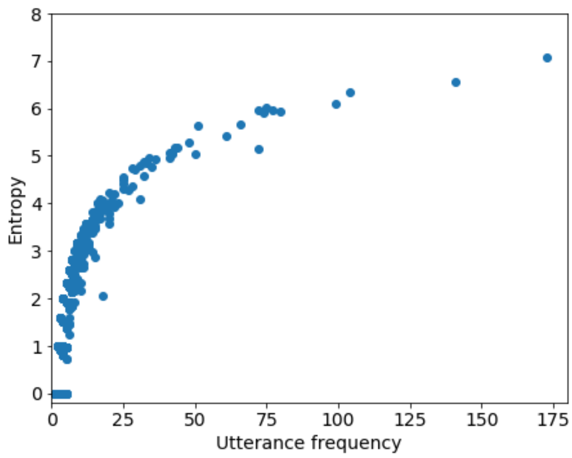

Figure 1: Entropy of source utterances (computed with IDENTITY) with respect to utterance frequency.

Entropy is clearly proportional to utterance frequency (Figure 1), but has a wide range of values among utterances of equal frequency. For example, utterances with a frequency of 3 can have entropies ranging from 0 to $\log_2 3 \approx 1.58$, the latter of which would be over our filtering threshold of 1 (see Section 5.1 for details on selecting thresholds). Since high-entropy utterances are relatively short, we also examined the relationship between entropy and utterance length (Figure 2). Given the relationship between frequency and entropy it comes as no surprise that longer utterances have lower entropy.

[3] http://yanran.li/dailydialog.html

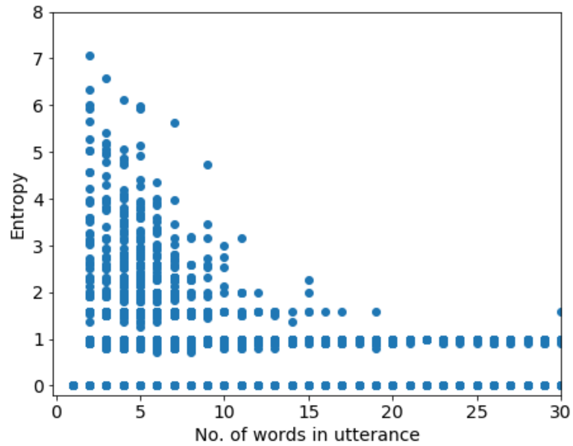

Figure 2: Entropy of source utterances (computed with IDENTITY) with respect to utterance length.

4.2 Clustering Results

Compared to IDENTITY, both SENT2VEC and AVG-EMBEDDING produce a much lower number of clusters with 0 entropy, but also a huge cluster with more than 5000 elements (the size of the second largest cluster is below 500), which we didn't filter since it clearly doesn't group utterances with similar meaning. Generally clusters were formed of similar utterances with the occasional exception of longer outlier utterances clustered together (instead of creating a separate cluster for each outlier), which can be attributed to the nature of the clustering algorithm. Overall, SENT2VEC appeared to produce better clusters than AVG-EMBEDDING, as reflected in the evaluation in Section 5.

We experimented with different bandwidth values[4] for the Mean Shift algorithm to produce clusters with as many elements as possible while also keeping the elements semantically similar. In an example cluster (Figure 3) we can see that the clustering was able to group together several variants of "How are you?", in particular those with different names. In general we noticed that both in the case of IDENTITY and the clustering methods, the utterances labeled with high entropy are the ones we wished to find: open-ended and without much context. A selection of high entropy clusters and utterances is presented in the Appendix.

[4] Bandwidth can be thought of as a radius in the latent space of utterance representations.

```
hi  an  .  how are you  ?
hi  craig  !  how are you  ?
hi  how are you  .  is alice there  ?
hi  !  how are you doing  ?
hi  francis  morning  !  how are you doing today  ?
hi  peter  !  how are you  ?
hi  randy  .  what are you doing right now  ?
hi  jane  .  how are you doing this morning  ?
hi  nancy  .  how are you doing  ?
hi  how are you doing  ?
hi  nancy . how are you doing  ?
hi steve  .  this is mike  .  what are you doing  ?
hi  how are you  ?
hi b  .  how are you  ?
hi  alex  .  how are you doing  ?
hi  !  how are you going  ?
hi mike  how are you doing  ?
hi  .  how can i help you  ?
hi  jack  !  how are you doing  ?
hi  carlos  .  what are you doing this afternoon  ?
hi  victor  .  how are you  ?
oh  yes  .  hi  how are you  ?
hi  tom  .  how have you been  ?
hi  bob  !  how are you doing  ?
hi  alice . how are you  ?
hi  brad  .  how are you today  ?
```

Figure 3: A cluster produced by SENT2VEC.

5 Experiments

In this section the model and parameter setups are presented along with 13 evaluation metrics. Some limitations of these metrics are discussed and a comparison of our filtering methods is given. We shall release all software required to replicate these results as well as to perform similar experiments on other datasets.

5.1 Model and Parameters

We use `transformer` (Vaswani et al., 2017) as our dialog model, an encoder-decoder architecture relying solely on attention mechanisms (Bahdanau et al., 2015), unlike standard `seq2seq` models. `transformer` has already been applied to a plethora of natural language processing tasks, including dialog modeling (Dinan et al., 2019; Mazare et al., 2018; Devlin et al., 2018). We used the official implementation[5] (see the Appendix for a report of all hyperparameters). The vocabulary was set to the most frequent 16 384 words, and the train / validation / test splits contained 71 517 / 9 027 / 9 318 examples, respectively.

Clustering and Filtering. For AVG-EMBEDDING fastText[6] embeddings were used. The bandwidth of Mean Shift was set to 0.7 and 3.5 for AVG-EMBEDDING and SENT2VEC, which produced 40 135 and 23 616 clusters, respectively. For IDENTITY we discarded utterance pairs whose target and/or source entropy is greater than 1, filtering many high-entropy utterances without shrinking the dataset too much (5-15%). The entropy threshold for each of the two clustering methods was chosen so that the amount of data that gets filtered is similar to IDENTITY (see the Appendix for exact figures). We also set a threshold for the maximum average utterance length (15 and 20 for AVG-EMBEDDING and SENT2VEC) in clusters that we considered for filtering, excluding outliers from the filtering process (see Section 4.2).

Training and Decoding. Word embeddings of size 512 were randomly initialized, batch size was set to 2048 tokens, and we used the Adam optimizer (Kingma and Ba, 2014). We experimented with different beam sizes (Graves, 2012), but greedy decoding performed better according to all our metrics, also observed previously (Asghar et al., 2017; Shao et al., 2017; Tandon et al., 2017).

5.2 Evaluation Metrics

As mentioned in Section 2, automatic evaluation of chatbots is an open research problem. In order to get as complete a picture as possible, we use 13 metrics that have been applied to dialog models over the past years. We briefly describe each metric and mention its past uses. BLEU and perplexity are omitted, because it has been shown that they are poor at measuring performance and don't correlate with human judgment (Liu et al., 2016). Even though we don't use these, we still have some metrics that compare model responses to targets, so the evaluation of this aspect of response quality is not omitted. It is also important that the following metrics assess different parts of response quality, thus models should be compared on the whole set of metrics. The first and simplest metric, response length ($|U|$), is widely used as a simple indicator of engagement (Serban et al., 2017b; Tandon et al., 2017; Baheti et al., 2018).

Word and utterance entropy. The per-word entropy $H_w = -\frac{1}{|U|}\sum_{w \in U} \log_2 p(w)$ of responses is measured to determine their non-genericness, where the probabilities are calculated based on frequencies observed in the training data (Serban et al., 2017b). We introduce the bigram version of this metric, to measure diversity at the bigram level as well. Utterance entropy is simply the

[5] https://github.com/tensorflow/tensor2tensor

[6] https://fasttext.cc/

product of H_w and $|U|$ (Serban et al., 2017b). We also report utterance entropy at the bigram level.

KL divergence. We use the KL divergence between model and ground truth (GT) response sets to measure how well a model can approximate the GT distribution of words. Specifically, we define distributions p_{gt} and p_m based on each set of responses and calculate the KL divergence $D_{KL} = \frac{1}{|U_{gt}|} \sum_{w \in U_{gt}} \log_2 \frac{p_{gt}(w)}{p_m(w)}$ for each GT response. The bigram version of this metric is also reported.

Embedding metrics. Embedding *average*, *extrema*, and *greedy* are widely used metrics introduced by Liu et al. (2016). *average* measures the cosine similarity between the average word embeddings of the response and the target. *extrema* constructs a representation by taking the greatest absolute value for each dimension among the word vectors in the response and target utterances, and measures the cosine similarity between them. Finally, *greedy* matches each response token to a target token (and vice versa) based on the cosine similarity between their embeddings, and averages the total score across all words. For word embeddings and average word embedding representations we used the same setup as in AVG-EMBEDDING.

Coherence. We measure the cosine similarity between pairs of input and response (Xu et al., 2018b). Although a coherence value of 1 would indicate that input and response are the same, generally a higher value seems better as model responses tend to have lower coherence than targets.

Distinct metrics. *Distinct-1* and *distinct-2* are widely used in the literature (Li et al., 2016a), measuring the ratio of unique unigrams/bigrams to the total number of unigrams/bigrams in a set of responses. However, they are very sensitive to the test data size, since increasing the number of examples in itself lowers their value. While the number of total words increases linearly, the number of unique words is limited by the vocabulary, and we found that the ratio decreases even in human data (a figure is included in the Appendix). It is therefore important to only compare distinct metrics computed on the same test data.

Normally metrics are computed at the validation loss minimum of a model, however in the case of chatbot models loss may not be a good indicator of response quality (Section 2), thus we also looked at how these 13 metrics progress during training.

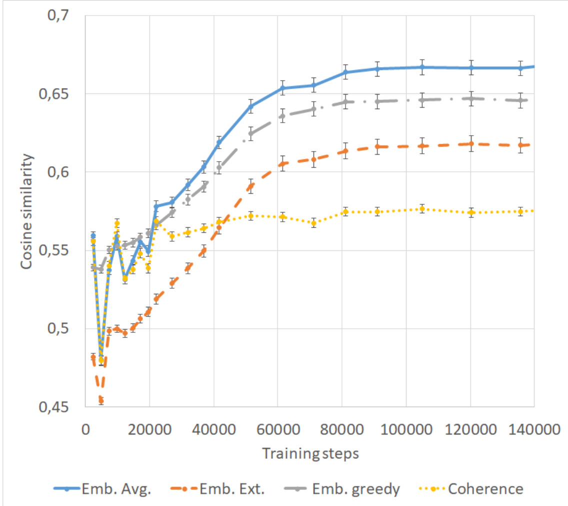

Figure 4: Embedding metrics and coherence (computed on validation data) as a function of the training evolution of `transformer` on unfiltered data.

Figure 4 shows how coherence and the 3 embedding metrics saturate after about 80-100k steps, and never decrease (we ran the training for 300k steps, roughly 640 epochs). All 13 metrics show a similar trend of increasing until 100k steps, and then stagnating (see the Appendix for more figures).

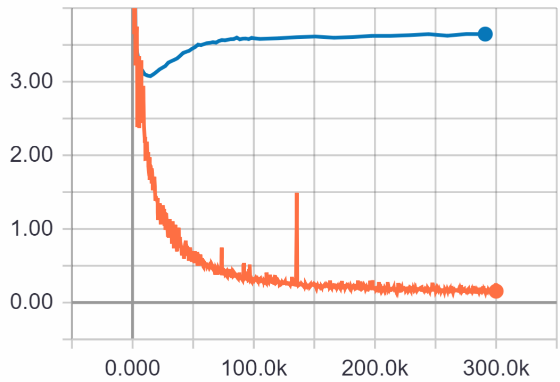

Figure 5: Training (orange/bottom) and validation (blue/top) loss progress with respect to the training evolution of `transformer` on unfiltered data.

In contrast, validation loss for the same training reaches its minimum after about 10-20k steps (Figure 5). This again suggests the inadequacy of the loss function, but it also questions the validity of these metrics, as they seem to favor a model that overfitted the training data, which we can assume after 640 epochs. This could be attributed to the fact that there are lots of identical inputs in train and test splits, because of the nature of dialog data. Most interesting are embedding metrics, since they show that even after overfitting responses do not get farther from targets. This is in line with other findings reporting that qualitatively responses are better after overfitting (Csáky, 2017; Tandon et al., 2017), however occasionally they also tend to be too specific and irrelevant. We leave it for future work to conduct human evaluation between non-overfitted and overfitted models to solidify these claims. In light of these issues we compare models both at the validation loss minimum and at an overfitted point where the metrics start stagnating (150 epochs).

5.3 Results

We compute metrics on the unfiltered test set to show that filtered trainings perform better even on utterances that would have been filtered from the training data. In Tables 7 and 9, TRF refers to the baseline `transformer` model trained on unfiltered data. The other 9 models are also `transformers`, but trained on the filtered datasets. The three methods for computing entropy are compared, *ID* refers to IDENTITY, *AE* to AVG-EMBEDDING, and *SC* to SENT2VEC. For each we compare the three filtering options: SOURCE-side, TARGET-side, or filtering BOTH sides, denoted by the initials in the tables. We highlight best results with bold (and those within a 95% confidence interval) in each column. * indicates the best result (and those within a 95% confidence interval) separately for each entropy computing method in order to compare between the 3 filtering options, but only if the result is statistically significant compared to the baseline. Tables with standard deviations can be found in the Appendix. The 13 metrics from left to right are: response length, unigram and bigram entropy, unigram and bigram utterance entropy, unigram and bigram KL divergence, embedding *average*, *extrema* and *greedy*, coherence, and finally, the *distinct-1* and *distinct-2* metrics (see Section 5.2).

Evaluating at the minimum validation loss (Table 7) clearly shows that models trained on data filtered by IDENTITY and SENT2VEC are better than the baseline. IDENTITY performs best among the three methods, surpassing the baseline on all but the *distinct-1* metric. SENT2VEC is a close second, getting higher values on fewer metrics than IDENTITY, but mostly improving on the baseline. Finally, AVG-EMBEDDING is inferior to the baseline, as it didn't produce clusters as meaningful as SENT2VEC, and thus produced a lower quality training set. It seems like filtering high entropy targets (both in the case of IDENTITY and SENT2VEC) is more beneficial than filtering sources, and BOTH falls mostly in the middle as expected, since it combines the two filtering types. By removing example responses that are boring and generic from the dataset the model learns to improve response quality. Finding such utterances is useful for a number of purposes, but earlier it has been done mainly manually (Li et al., 2016d; Shen et al., 2017), whereas our method can automatically find them based on entropy.

Every value is higher after 150 epochs of training than at the validation loss minimum (Table 9). The most striking change is in the unigram KL divergence, which is now an order of magnitude lower. IDENTITY still performs best, falling behind the baseline on only the two distinct metrics. Interestingly this time BOTH filtering was better than the TARGET filtering. SENT2VEC still mostly improves the baseline and AVG-EMBEDDING now also performs better or at least as good as the baseline on most metrics. In the **GT** row of Table 9 we can see the metrics computed on the ground truth responses. In some cases the best performing model gets quite close to the ground truth performance. In the **RT** row we show the quality of randomly selected responses from the train set. On metrics that evaluate utterances without context (i.e. entropy, divergence, distinct), **RT** achieves similar values as the ground truth, which is expected. However, on all embedding metrics and coherence, **RT** scores are significantly worse than those of any model evaluated.

Computing the unigram and bigram KL divergence with a uniform distribution instead of the model yields a value of 4.35 and 1.87, respectively. Thus, all models learned a much better distribution, suggesting that this is indeed a useful metric. We believe the main reason that clustering methods perform worse than IDENTITY is that clustering adds some noise to the filtering process. This could be remedied by filtering only utterances instead of whole clusters, thus combining IDENTITY and the clustering methods. In this scenario, the entropy of individual utterances is computed based on the clustered data. The intuition behind this approach would be that the noise in the clusters based on which we compute entropy

		$\lvert U \rvert$	H_w^u	H_w^b	H_u^u	H_u^b	D_{KL}^u	D_{KL}^b	AVG	EXT	GRE	COH	d-1	d-2
TRF		8.61	7.30	12.2	63.6	92.8	0.330	0.850	0.540	0.497	0.552	0.538	**0.0290**	0.149
ID	B	9.79	7.44	12.3	71.9	105	0.315	0.765	0.559	**0.506***	0.555*	0.572	0.0247	0.138
	T	10.9*	**7.67***	**12.7***	**83.2***	**121***	**0.286***	**0.719***	**0.570***	**0.507***	0.554	**0.584***	0.0266	**0.150***
	S	9.36	7.19	11.9	66.4	98.2	0.462	1.08	0.540	0.495	0.553	0.538	0.0262	0.130
AE	B	7.93	7.25	12.0	57.7	82.9	0.447	1.05	0.524	0.486	0.548	0.524	0.0283	0.132
	T	8.57	7.26	12.1	61.4	90.3	0.425	1.12	0.526	0.492	0.548	0.529	0.0236	0.115
	S	9.04*	7.21	11.9	65.1*	94.8*	0.496	1.16	0.536	0.490	0.548	0.538	0.0232	0.109
SC	B	10.0	7.40	12.3	72.6	108	0.383	0.974	0.544	0.497	0.549	0.550	0.0257	0.131
	T	**11.2***	7.49*	12.4*	**82.2***	**122***	0.391	0.967	0.565*	0.500*	0.552	0.572*	0.0250	0.132
	S	**11.1***	7.15	11.9	74.4	114	0.534	1.27	0.546	0.501*	**0.560***	0.544	0.0213	0.102

Table 1: Metrics computed at the minimum of the validation loss on the unfiltered test set.

		$\lvert U \rvert$	H_w^u	H_w^b	H_u^u	H_u^b	D_{KL}^u	D_{KL}^b	AVG	EXT	GRE	COH	d-1	d-2
TRF		11.5	7.98	**13.4**	95.1	142	0.0360	0.182	0.655	**0.607**	**0.640**	0.567	**0.0465**	**0.297**
ID	B	**13.1***	**8.08***	**13.6**	**107***	**162***	0.0473	0.210	**0.668***	**0.608**	**0.638**	**0.598***	0.0410	0.275
	T	12.2	8.04	**13.6**	99.9	150	**0.0335***	**0.181***	**0.665***	**0.610**	**0.640**	0.589	0.0438	0.289
	S	12.3	7.99	**13.5**	101	153	0.0406	0.187	0.662	**0.610**	**0.641**	0.578	0.0444	0.286
AE	B	11.9	7.98	**13.5**	97.6	147	0.0395	0.197	0.649	0.600	0.628	0.574	0.0434	0.286
	T	12.5*	7.99	**13.5**	102*	155*	0.0436	0.204	0.656	0.602	0.634	0.580*	0.0423	0.279
	S	12.1	7.93	**13.4**	98.6	148	0.0368	0.186	0.658	0.605	**0.636**	0.578*	0.0425	0.278
SC	B	12.8	**8.07***	**13.6**	105	159	0.0461	0.209	0.655	0.600	0.629	0.583*	0.0435	0.282
	T	**13.0***	**8.06***	**13.6**	**107***	**162***	0.0477	0.215	0.657	0.602	0.632	0.585*	0.0425	0.279
	S	12.1	7.96	**13.4**	99.5	150	0.0353*	0.183	0.657	**0.606**	**0.638**	0.576	0.0443	0.286
RT		13.5	8.40	14.2	116	177	0.0300	0.151	0.531	0.452	0.481	0.530	0.0577	0.379
GT		14.1	8.39	13.9	122	165	0	0	1	1	1	0.602	0.0488	0.362

Table 2: Metrics computed on the unfiltered test set after 150 epochs of training.

Input	Response
your starting salary is 2500 yuan a month and after you become a permanent employee it will be higher .	**BASE:** i can tell you what is the best way to find a job . **BASE-O:** do you know what it is ? **TARGET:** i 'm very interested in the position . **TARGET-O:** that 's very nice of you . i 'll have to think about it .
you can greatly improve your grades by always reflecting on what you can improve on .	**BASE:** i want to be a great singer . **BASE-O:** i really appreciate it . thanks . **TARGET:** i think i am experienced in that position . i think i would like to make a good impression . **TARGET-O:** i always liked it . thank you .
umm i think i would be hard to find something in that range in this area . you know the environment in the south district is the nicest .	**BASE:** what about the kitchen ? **BASE-O:** what about the kitchen ? **TARGET:** what about the kitchen ? **TARGET-O:** what about the kitchen ?

Table 3: Three example inputs and responses. BASE is the model trained on unfiltered data, and TARGET is the model trained on IDENTITY, TARGET filtered data. We mark with O models evaluated at an overfitted point.

is less harmful than the noise in clusters which we consider for filtering. Finally, Table 3 shows responses from the baseline and the best performing model to 3 randomly selected inputs from the test set (which we made sure are not present in the training set) to show that training on filtered data does not degrade response quality. We show more example responses in the Appendix.

6 Conclusion

We proposed a simple unsupervised entropy-based approach that can be applied to any conversational dataset for filtering generic sources/targets that cause "confusion" during the learning of open-domain dialog models. We compared various setups in an extensive quantitative evaluation, and showed that IDENTITY and SENT2VEC perform best, usually achieving the largest improvement over the baseline through only TARGET filtering.

Some limitations of current automatic metrics and the loss function have also been shown, by examining their behavior on random data and with overfitting.

For future work we wish to explore several objectives. As mentioned in Section 5.3, we want to extend our clustering experiments combining the ideas behind IDENTITY and the clustering methods to make them more robust to noise. We also want to experiment with different datasets (Danescu-Niculescu-Mizil and Lee, 2011; Zhang et al., 2018b), other models that can also handle dialog history, like HRED (Serban et al., 2016), or VHCR (Park et al., 2018), and with combining coherence-based filtering (Xu et al., 2018b) with our method. Moreover, we wish to gain better qualitative insights by also conducting human evaluation. Finally, we believe our method is general enough that it could also be applied to datasets in other similar NLP tasks, such as machine translation, which could open another interesting line of future research.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*.

Nabiha Asghar, Pascal Poupart, Xin Jiang, and Hang Li. 2017. Deep active learning for dialogue generation. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 78–83. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR 2015)*.

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.

Kyunghyun Cho, Bart van Merriëenboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar.

Richárd Csáky. 2017. Deep learning based chatbot models. Technical report, Budapest University of Technology and Economics, Budapest.

Cristian Danescu-Niculescu-Mizil and Lillian Lee. 2011. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*, pages 76–87. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*.

Hao Fang, Hao Cheng, Maarten Sap, Elizabeth Clark, Ari Holtzman, Yejin Choi, Noah A. Smith, and Mari Ostendorf. 2018. Sounding board: A user-centric and content-driven social chatbot. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 96–100. Association for Computational Linguistics.

Keinosuke Fukunaga and Larry Hostetler. 1975. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, 21(1):32–40.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *Thirty-Second AAAI Conference on Artificial Intelligence*. Association for the Advancement of Artificial Intelligence.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. In *Representation Learning Workshop, ICML 2012*, Edinburgh, Scotland.

Xiaodong Gu, Kyunghyun Cho, Jung-Woo Ha, and Sunghun Kim. 2019. DialogWAE: Multimodal response generation with conditional wasserstein auto-encoder. In *International Conference on Learning Representations*.

Chaitanya K Joshi, Fei Mi, and Boi Faltings. 2017. Personalization in goal-oriented dialog. *arXiv preprint arXiv:1706.07503*.

Kirthevasan Kandasamy, Yoram Bachrach, Ryota Tomioka, Daniel Tarlow, and David Carter. 2017. Batch policy gradient methods for improving neural conversation models. *arXiv preprint arXiv:1702.03334*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ben Krause, Marco Damonte, Mihai Dobre, Daniel Duma, Joachim Fainberg, Federico Fancellu, Emmanuel Kahembwe, Jianpeng Cheng, and Bonnie Webber. 2017. Edina: Building an open domain socialbot with self-dialogues. In *1st Proceedings of Alexa Prize (Alexa Prize 2017)*.

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*.

Mike Lewis, Denis Yarats, Yann N Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? end-to-end learning for negotiation dialogues. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2443–2453. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proceedings of NAACL-HLT 2016*, pages 110–119. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan. 2016b. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 994–1003. Association for Computational Linguistics.

Jiwei Li, Alexander H Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2016c. Dialogue learning with human-in-the-loop. *arXiv preprint arXiv:1611.09823*.

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

Jiwei Li, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. 2017a. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169. Association for Computational Linguistics.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017b. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the The 8th International Joint Conference on Natural Language Processing*, pages 986–995. AFNLP.

Zachary Lipton, Xiujun Li, Jianfeng Gao, Lihong Li, Faisal Ahmed, and Li Deng. 2018. Bbq-networks: Efficient exploration in deep reinforcement learning for task-oriented dialogue systems. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*. Association for the Advancement of Artificial Intelligence.

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. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132. Association for Computational Linguistics.

Huiting Liu, Tao Lin, Hanfei Sun, Weijian Lin, Chih-Wei Chang, Teng Zhong, and Alexander Rudnicky. 2017. Rubystar: A non-task-oriented mixture model dialog system. In *1st Proceedings of Alexa Prize (Alexa Prize 2017)*.

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.

Oswaldo Ludwig. 2017. End-to-end adversarial learning for generative conversational agents. *arXiv preprint arXiv:1711.10122*.

Pierre-Emmanuel Mazare, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training millions of personalized dialogue agents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2775–2779. Association for Computational Linguistics.

Kaixiang Mo, Yu Zhang, Qiang Yang, and Pascale Fung. 2017. Fine grained knowledge transfer for personalized task-oriented dialogue systems. *arXiv preprint arXiv:1711.04079*.

Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. 2016. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3349–3358. The COLING 2016 Organizing Committee.

Oluwatobi Olabiyi, Alan Salimov, Anish Khazane, and Erik Mueller. 2018. Multi-turn dialogue response generation in an adversarial learning framework. *arXiv preprint arXiv:1805.11752*.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 528–540. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the*

40th Annual Meeting of the Association for Computational Linguistics, Philadelphia.

Yookoon Park, Jaemin Cho, and Gunhee Kim. 2018. A hierarchical latent structure for variational conversation modeling. In *Proceedings of NAACL-HLT 2018*, pages 1792–1801. Association for Computational Linguistics.

Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, et al. 2018. Conversational ai: The science behind the alexa prize. *arXiv preprint arXiv:1801.03604*.

Joao Sedoc, Daphne Ippolito, Arun Kirubarajan, Jai Thirani, Lyle Ungar, and Chris Callison-Burch. 2018. Chateval: A tool for the systematic evaluation of chatbots. In *Proceedings of the Workshop on Intelligent Interactive Systems and Language Generation (2IS&NLG)*, pages 42–44. Association for Computational Linguistics.

Iulian V Serban, Chinnadhurai Sankar, Mathieu Germain, Saizheng Zhang, Zhouhan Lin, Sandeep Subramanian, Taesup Kim, Michael Pieper, Sarath Chandar, Nan Rosemary Ke, et al. 2017a. A deep reinforcement learning chatbot. *arXiv preprint arXiv:1709.02349*.

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

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C Courville, and Yoshua Bengio. 2017b. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*. Association for the Advancement of Artificial Intelligence.

Igor Shalyminov, Ondřej Dušek, and Oliver Lemon. 2018. Neural response ranking for social conversation: A data-efficient approach. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 1–8. Association for Computational Linguistics.

Yuanlong Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 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*, pages 2210–2219. Association for Computational Linguistics.

Xiaoyu Shen, Hui Su, Wenjie Li, and Dietrich Klakow. 2018a. Nexus network: Connecting the preceding and the following in dialogue generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4316–4327. Association for Computational Linguistics.

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.

Xiaoyu Shen, Hui Su, Shuzi Niu, and Vera Demberg. 2018b. Improving variational encoder-decoders in dialogue generation. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*. Association for the Advancement of Artificial Intelligence.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proc. NIPS*, pages 3104–3112, Montreal, CA.

Shubhangi Tandon, Ryan Bauer, et al. 2017. A dual encoder sequence to sequence model for open-domain dialogue modeling. *arXiv preprint arXiv:1710.10520*.

Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. 2018. Ruber: An unsupervised method for automatic evaluation of open-domain dialog systems. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*. Association for the Advancement of Artificial Intelligence.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Oriol Vinyals and Quoc V. Le. 2015. A neural conversational model. In *Proceedings of the 31st International Conference on Machine Learning*.

Yansen Wang, Chenyi Liu, Minlie Huang, and Liqiang Nie. 2018. Learning to ask questions in open-domain conversational systems with typed decoders. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, pages 2193–2203. Association for Computational Linguistics.

Bolin Wei, Shuai Lu, Lili Mou, Hao Zhou, Pascal Poupart, Ge Li, and Zhi Jin. 2017. Why do neural dialog systems generate short and meaningless replies? a comparison between dialog and translation. *arXiv preprint arXiv:1712.02250*.

Marek Wojciechowski and Maciej Zakrzewicz. 2002. Dataset filtering techniques in constraint-based frequent pattern mining. In *Pattern detection and discovery*, pages 77–91. Springer.

Bowen Wu, Nan Jiang, Zhifeng Gao, Suke Li, Wenge Rong, and Baoxun Wang. 2018. Why do neural response generation models prefer universal replies? *arXiv preprint arXiv:1808.09187*.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*. Association for the Advancement of Artificial Intelligence.

Chen Xing, Yu Wu, Wei Wu, Yalou Huang, and Ming Zhou. 2018. Hierarchical recurrent attention network for response generation. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*. Association for the Advancement of Artificial Intelligence.

Yujie Xing and Raquel Fernández. 2018. Automatic evaluation of neural personality-based chatbots. In *Proceedings of The 11th International Natural Language Generation Conference*, pages 189–194. Association for Computational Linguistics.

Can Xu, Wei Wu, and Yu Wu. 2018a. Towards explainable and controllable open domain dialogue generation with dialogue acts. *arXiv preprint arXiv:1807.07255*.

Xinnuo Xu, Ondřej Dušek, Ioannis Konstas, and Verena Rieser. 2018b. Better conversations by modeling, filtering, and optimizing for coherence and diversity. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3981–3991. Association for Computational Linguistics.

Hainan Zhang, Yanyan Lan, Jiafeng Guo, Jun Xu, and Xueqi Cheng. 2018a. Reinforcing coherence for sequence to sequence model in dialogue generation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*, pages 4567–4573.

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 (Long Papers)*, pages 2204–2213. Association for Computational Linguistics.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018c. Generating informative and diverse conversational responses via adversarial information maximization. In *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*.

Tiancheng Zhao, Kyusong Lee, and Maxine Eskenazi. 2018. Unsupervised discrete sentence representation learning for interpretable neural dialog generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, pages 1098–1107. Association for Computational Linguistics.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664. Association for Computational Linguistics.

Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*. Association for the Advancement of Artificial Intelligence.

Wenya Zhu, Kaixiang Mo, Yu Zhang, Zhangbin Zhu, Xuezheng Peng, and Qiang Yang. 2017. Flexible end-to-end dialogue system for knowledge grounded conversation. *arXiv preprint arXiv:1709.04264*.

Appendix

High Entropy Clusters and Utterances

Top 20 high entropy utterances

Utterance	Frequency	Entropy
yes .	173	7.06
thank you .	141	6.57
why ?	104	6.33
here you are .	99	6.10
ok .	75	6.00
what do you mean ?	77	5.97
may i help you ?	72	5.96
can i help you ?	80	5.93
really ?	74	5.91
sure .	66	5.66
what can i do for you ?	51	5.63
why not ?	61	5.42
what ?	48	5.27
what happened ?	44	5.18
anything else ?	43	5.17
thank you very much .	72	5.14
what is it ?	41	5.06
i see .	42	5.05
no .	42	5.04
thanks .	50	5.03

Table 4: Top 20 source utterances sorted by entropy. The entropy was calculated with IDENTITY.

High Entropy Clusters

```
Center: coffee  ?  i don  t honestly like that kind of stuff  .
Entropy: 5.885753989955374
Size: 138
Elements:
here you are  .
here you are  .  have a nice stay here  .
here they are  .
you are kidding  .
of course  .  here you are  .
here you are  madam . all these are sixteens  .
we are here  .
here we are  .  this is wangfujing street  .
here you are  .  you left the medicine here  .
certainly  here you are  .
of course . here you are  .
sure  here you are  .
here you are . you can try them on  .
here you are . it s very attractive  .
here we are  .
surely of course . here you are  .
of course  here you are  .
you are late  .
thank you  .  here you are  .
here you are  madam  .  all these are sixteens  .
```

Figure 6: A high entropy cluster.

```
Center: come on  you can at least try a little  besides your cigarette  .
Entropy: 4.959251313559618
Size: 140
Elements:
thank you very much for your kindness  .
yes  please  .  thank you very much  .
sure . thank you very much  .
thank you very much  .  it  s very kind of you  .
okay  .  thank you very much  .
thank you very much  .
you are so kind  !  thank you very much  .
yes . thank you very much  .
thank you very much  .  see you tomorrow afternoon  .
i love flowers you know  .  thank you very much  .
yes  thank you very much  .
i see  .  thank you very much  .
thank you very much  .   take the pills  .
well  thank you very much  .
thank you very much  .  are you here alone  ?
here it is  .  and thank you very much  .
i understand  .  thank you very much  !
oh  thank you very much  .
fine  thank you very much  .
ok  thank you very much  .
oh  thank you so much  .
i  ll bring the card  .  thank you very much  .
all right  .  thank you very much  .
ok  i see  .  thank you very much  .
quite well  thank you  .
i know  .  thank you very much  .
i love flowers  you know  .  thank you very much  .
very well  thank you  .
thank you very much  .  byebye  .
oh  well thank you very much  .
thank goodness  .  it is still there  .  thank you very much  .
thank you so much  .
thank you very much  !
thank you very much  doctor  .
okay  sir  here you are  .  thank you very much  .
yes  thank you so much  .
fantastic  .  thank you very much  .
thank you very much  mr green  .
well  thank you  .
```

Figure 7: A high entropy cluster.

```
Center: i 'm not sure  .  but i 'll get a table ready as fast as i can  .
Entropy: 4.638892533270529
Size: 57
Elements:
yes  follow me  .  here it is  .
oh  yes  .  here it is  .
yes  here this is  .
oh  .  yes  .  it is
yes  .  here it is  .
oh  yes  it is  .
yes  we are  .
yes  it has  .
oh  yes . here it is  .
yes  it 's  167  .
yes  they are  .
yes  sir  .  here it is  .
yes  she is  .
yes  it is  .  it 's brilliant  .
yes  here it is  .
yes  it is  .
yes  he is  .
yes  it would be  .
```

Figure 8: A high entropy cluster.

Model and Other Parameters

Type	Threshold	SOURCE	TARGET	BOTH
ID	1	5.64%	6.98%	12.24%
AE	3.5	5.39%	7.06%	12.0%
SC	3.5	6.53%	8.45%	14.3%

Table 5: Entropy threshold and amount of data filtered in the 3 filtering scenarios, for the three entropy computing methods. ID stands for IDENTITY, AE stands for AVG-EMBEDDING clustering, and SC for SENT2VEC clustering.

Name	Value
Hidden size	512
Number of hidden layers	6
Label smoothing	0.1
Filter size	2048
Number of attention heads	8
Layer dropout	0.2
Relu dropout	0.1
Attention dropout	0.1
Learning rate	0.2
Learning rate warmup steps	8000

Table 6: Transformer hyperparameters.

Evaluation Metrics and Results

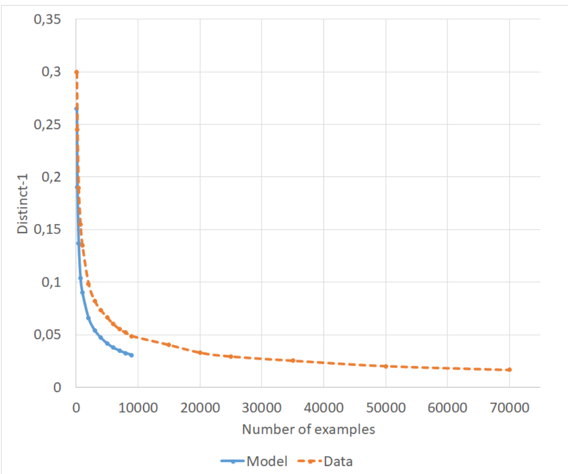

Figure 9: Distinct-1 metric with respect to number of test examples. Model responses were evaluated on 9000 examples only, since the rest were training examples.

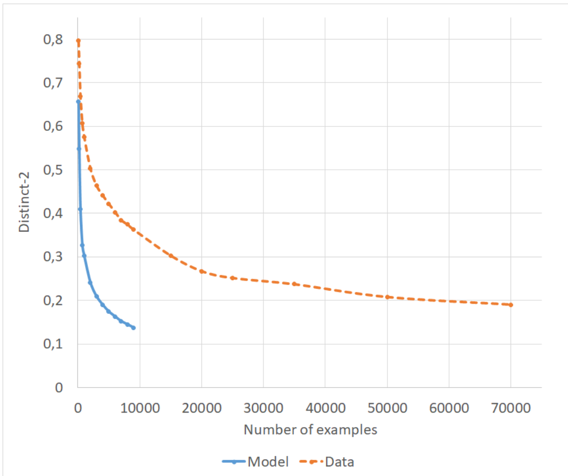

Figure 10: Distinct-2 metric with respect to number of test examples. Model responses were evaluated on 9000 examples only, since the rest were training examples.

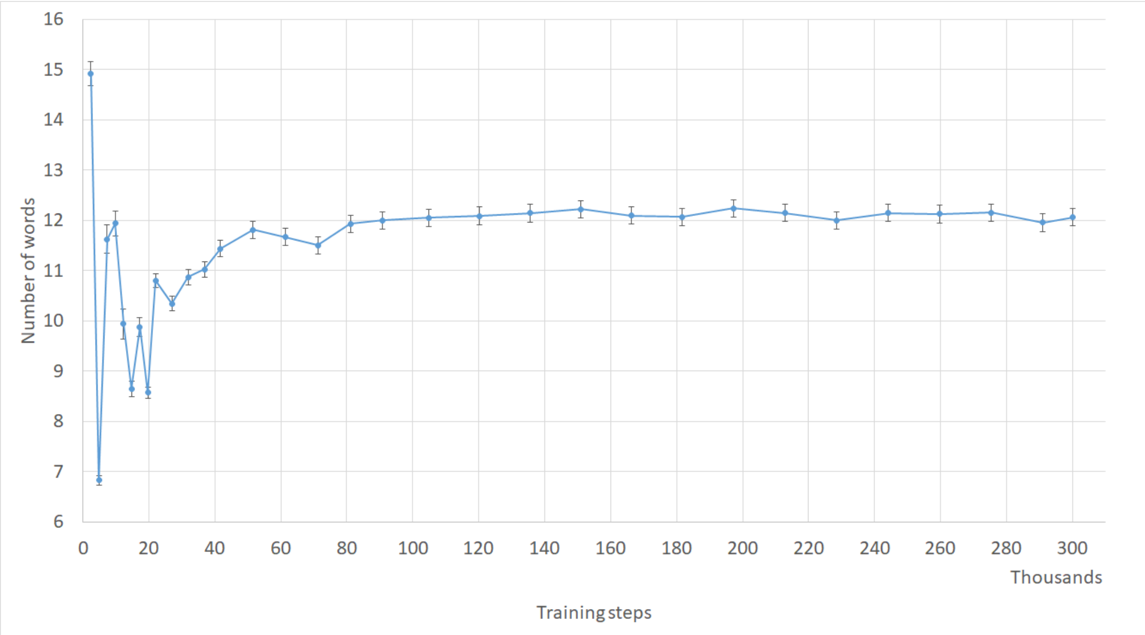

Figure 11: Average length of responses (computed on the validation set) with respect to the number of training steps of the `transformer` trained on unfiltered data.

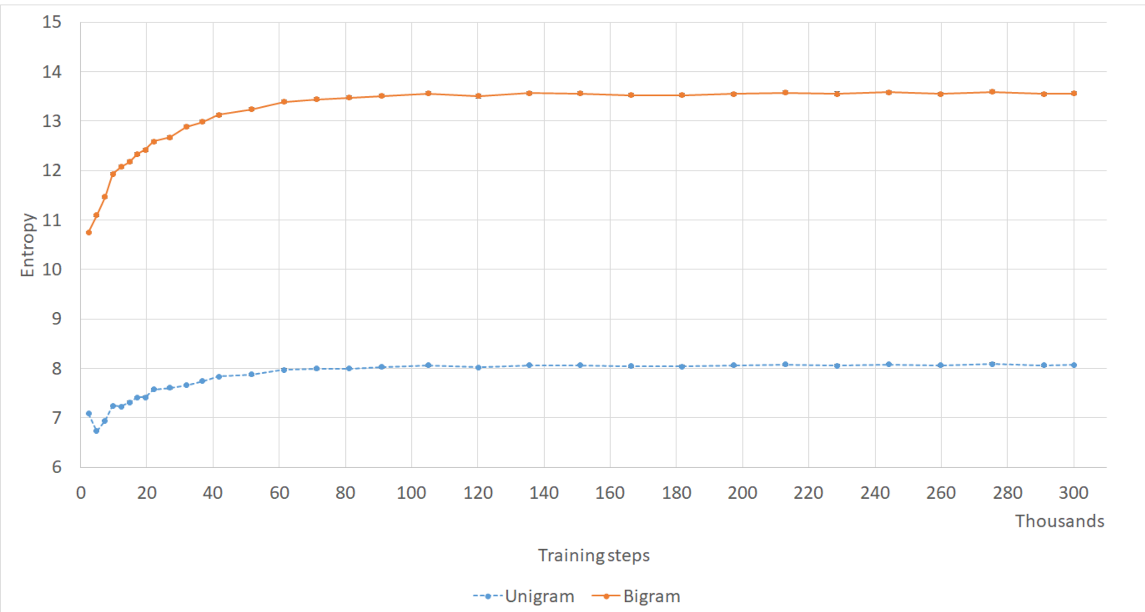

Figure 12: Word entropy of responses (computed on the validation set) with respect to the number of training steps of the `transformer` trained on unfiltered data.

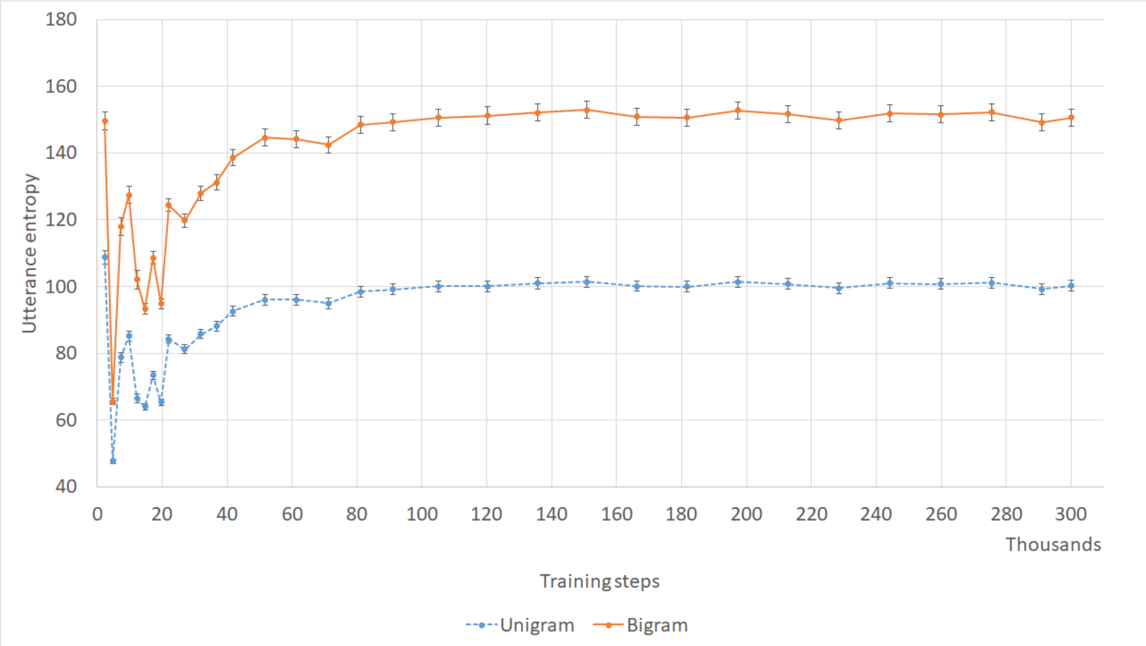

Figure 13: Utterance entropy of responses (computed on the validation set) with respect to the number of training steps of the `transformer` trained on unfiltered data.

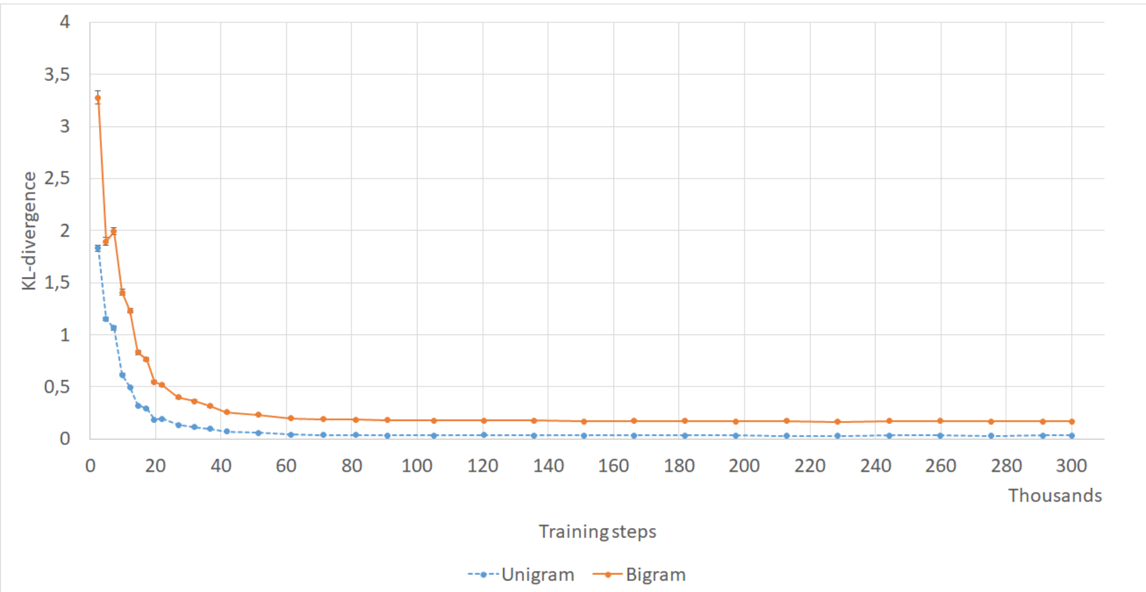

Figure 14: KL divergence of responses (computed on the validation set) with respect to the number of training steps of the `transformer` trained on unfiltered data.

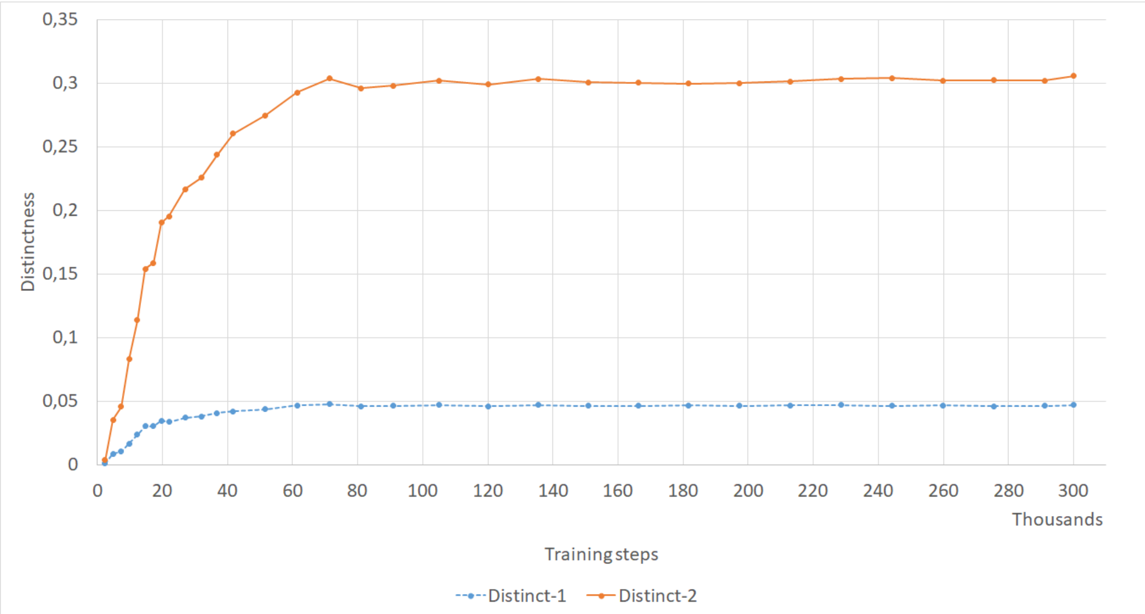

Figure 15: Distinct-1 and distinct-2 metrics (computed on the validation set) with respect to the number of training steps of the `transformer` trained on unfiltered data.

	$\|U\|$	$H_{w,u}$	$H_{w,b}$	$H_{u,u}$	$H_{u,b}$	$D_{KL,u}$	$D_{KL,b}$
TRF	8.61 (7.79)	7.30 (0.950)	12.2 (1.55)	63.6 (46.2)	92.8 (81.5)	0.330 (0.458)	0.850 (0.927)
ID							
B	9.79 (6.98)	7.44 (0.829)	12.3 (1.33)	71.9 (40.2)	105 (70.2)	0.315 (0.450)	0.765 (0.851)
T	10.9 (6.61)*	**7.67 (0.816)***	**12.7 (1.30)***	**83.2 (49.8)***	**121 (73.4)***	**0.286 (0.395)***	**0.719 (0.789)***
S	9.36 (9.99)	7.19 (0.985)	11.9 (1.51)	66.4 (50.5)	98.2 (96.9)	0.462 (0.531)	1.08 (1.00)
AE							
B	7.93 (7.28)	7.25 (0.976)	12.0 (1.58)	57.7 (40.5)	82.9 (73.8)	0.447 (0.538)	1.05 (1.07)
TT	8.57 (8.65)	7.26 (0.970)	12.1 (1.55)	61.4 (43.1)	90.3 (83.0)	0.425 (0.511)	1.12 (1.11)
S	9.04 (8.20)*	7.21 (0.894)	11.9 (1.44)	65.1 (47.3)*	94.8 (82.0)*	0.496 (0.551)	1.16 (1.09)
SC							
B	10.0 (9.60)	7.40 (0.924)	12.3 (1.44)	72.6 (49.9)	108 (94.6)	0.383 (0.494)	0.974 (0.988)
T	**11.2 (8.81)***	7.49 (0.876)*	12.4 (1.38)*	**82.2 (47.6)***	**122 (88.1)***	0.391 (0.473)	0.967 (0.937)
S	**11.1 (14.7)***	7.15 (1.06)	11.9 (1.57)	74.4 (63.6)	114 (133)	0.534 (0.554)	1.27 (1.02)

Table 7: Metrics computed at the minimum of the validation loss for each model on the unfiltered test set. The 7 metrics from left to right are: response length, unigram and bigram word entropy, unigram and bigram utterance entropy, and unigram and bigram KL divergence.

	Average	Extrema	Greedy	Coherence	d-1	d-2
TRF	0.540 (0.147)	0.497 (0.131)	0.552 (0.124)	0.538 (0.142)	**0.0290**	0.149
ID.						
B	0.559 (0.139)	**0.506 (0.125)***	0.555 (0.119)*	0.572 (0.139)	0.0247	0.138
T	**0.570 (0.140)***	**0.507 (0.126)***	0.554 (0.119)	**0.584 (0.140)***	0.0266	**0.150***
S	0.540 (0.144)	0.495 (0.128)	0.553 (0.121)	0.538 (0.143)	0.0262	0.130
AE						
B	0.524 (0.144)	0.486 (0.126)	0.548 (0.120)	0.524 (0.145)	0.0283	0.132
T	0.526 (0.144)	0.492 (0.126)	0.548 (0.118)	0.529 (0.146)	0.0236	0.115
S	0.536 (0.141)	0.490 (0.126)	0.548 (0.119)	0.538 (0.143)	0.0232	0.109
SC						
B	0.544 (0.141)	0.497 (0.122)	0.549 (0.116)	0.550 (0.137)	0.0257	0.131
T	0.565 (0.137)*	0.500 (0.122)*	0.552 (0.115)	0.572 (0.135)*	0.0250	0.132
S	0.546 (0.143)	0.501 (0.128)*	**0.560 (0.122)***	0.544 (0.141)	0.0213	0.102

Table 8: Metrics computed at the minimum of the validation loss for each model on the unfiltered test set. The first 3 metrics are: embedding average, extrema and greedy. Next is the cosine similarity between input and response utterance (coherence), and finally the distinct-1 and distinct-2 metrics.

	$\|U\|$	$H_{w,u}$	$H_{w,b}$	$H_{u,u}$	$H_{u,b}$	$D_{KL,u}$	$D_{KL,b}$
TRF	11.5 (8.45)	7.98 (1.06)	13.4 (1.75)	95.1 (77.2)	142 (122)	0.0360 (0.167)	0.182 (0.392)
ID							
B	**13.1 (8.05)***	**8.08 (0.934)***	13.6 (1.54)	**107 (72.9)***	**162 (116)***	0.0473 (0.169)	0.210 (0.397)
T	12.2 (7.70)	8.04 (0.956)	13.6 (1.57)	99.9 (70.4)	150 (112)	**0.0335 (0.165)***	**0.181 (0.380)***
S	12.3 (8.55)	7.99 (1.02)	13.5 (1.66)	101 (77.6)	153 (124)	0.0406 (0.170)	0.187 (0.387)
AE							
B	11.9 (8.07)	7.98 (1.01)	13.5 (1.68)	97.6 (73.5)	147 (117)	0.0395 (0.168)	0.197 (0.394)
T	12.5 (8.40)*	7.99 (1.01)	13.5 (1.64)	102 (76.0)*	155 (121)*	0.0436 (0.171)	0.204 (0.404)
S	12.1 (8.34)	7.93 (1.01)	13.4 (1.68)	98.6 (75.5)	148 (120)	0.0368 (0.181)	0.186 (0.410)
SC							
B	12.8 (8.32)	**8.07 (0.971)***	13.6 (1.58)	105 (75.3)	159 (120)	0.0461 (0.164)	0.209 (0.401)
T	**13.0 (8.48)***	**8.06 (0.965)***	13.6 (1.57)	**107 (76.9)***	**162 (123)***	0.0477 (0.166)	0.215 (0.403)
S	12.1 (8.65)	7.96 (1.04)	13.4 (1.69)	99.5 (78.4)	150 (125)	0.0353 (0.169)*	0.183 (0.403)
RT	13.5 (9.78)	8.40 (1.09)	14.2 (1.68)	116 (91.8)	177 (146)	0.0300 (0.133)	0.151 (0.343)
GT	14.1 (10.9)	8.39 (1.05)	13.9 (1.56)	122 (102)	165 (141)	0	0

Table 9: Metrics computed on the unfiltered test set after 150 epochs of training. The 7 metrics from left to right are: response length, unigram and bigram word entropy, unigram and bigram utterance entropy, and unigram and bigram KL divergence. **GT** refers to the ground truth test set responses. **RT** refers to randomly selected responses from the train set.

	Average	Extrema	Greedy	Coherence	d-1	d-2
TRF	0.655 (0.225)	**0.607 (0.237)**	**0.640 (0.220)**	0.567 (0.144)	**0.0465**	**0.297**
ID						
B	**0.668 (0.213)***	**0.608 (0.230)**	**0.638 (0.215)**	**0.598 (0.139)***	0.0410	0.275
T	**0.665 (0.217)***	**0.610 (0.233)**	**0.640 (0.217)**	0.589 (0.140)	0.0438	0.289
S	0.662 (0.222)	**0.610 (0.237)**	**0.641 (0.221)**	0.578 (0.140)	0.0444	0.286
AE						
B	0.649 (0.219)	0.600 (0.229)	0.628 (0.215)	0.574 (0.141)	0.0434	0.286
T	0.656 (0.217)	0.602 (0.230)	0.634 (0.214)	0.580 (0.140)*	0.0423	0.279
S	0.658 (0.216)	0.605 (0.231)	**0.636 (0.214)**	0.578 (0.139)*	0.0425	0.278
SC						
B	0.655 (0.216)	0.600 (0.230)	0.629 (0.215)	0.583 (0.139)*	0.0435	0.282
T	0.657 (0.215)	0.602 (0.229)	0.632 (0.214)	0.585 (0.137)*	0.0425	0.279
S	0.657 (0.221)	**0.606 (0.235)**	**0.638 (0.218)**	0.576 (0.142)	0.0443	0.286
RT	0.531 (0.123)	0.452 (0.0990)	0.481 (0.0856)	0.530 (0.122)	0.0577	0.379
GT	1	1	1	0.602 (0.146)	0.0488	0.362

Table 10: Metrics computed on the unfiltered test set after 150 epochs of training. The first 3 metrics are: embedding average, extrema and greedy. Next is the cosine similarity between input and response utterance (coherence), and finally the distinct-1 and distinct-2 metrics. **GT** refers to the ground truth test set responses. **RT** refers to randomly selected responses from the train set.

Input	Unfiltered	Unfiltered-Overfitted	IDENTITY TARGET filtered	IDENTITY TARGET filtered-overfitted
some high speed chase scene i think .	i 'm not sure .	but many things are went wrong !	i 'm so sorry .	enough !
how long is this bus ride ?	about twelve hours .	it 's about fortyfive minutes to an hour .	it 's about 20 minutes .	it 's only about fortyfive minutes to an hour .
is that yellow bus it ?	no it 's not far . it 's a bit cold .	yeah . there s a new one in america .	no it 's a yellow one .	no it 's not very expensive these days .
hi mary ! what do you like to do in your spare time ?	i like playing basketball .	well i like to play the piano .	i like playing basketball .	i like playing chess .
let me see . we can visit the country . we have n't done that for such a long time .	i hope so .	yes there are many famous paintings on show . i hope it 's the nicest city in the united states .	i 'm glad to hear that .	where are your parents ?
thirty u . s . dollars . . . and it 's only been used for six months .	that 's fine .	i see . is it a long time to come to the station ?	how many months did you get there ?	what 're two weeks ?
five hundred dollars altogether .	here you are .	here you are .	here is your bill .	here you are . thank you byebye .
look at this bridge . we will soon be crossing the thames .	yes i agree .	oh yeah .	how far is it ?	well there 's a spanish teacher . so i 'll often go back to the spanish museum .
is the furniture crafted well ?	yes it is .	yes it 's fully furnished .	yes the furniture is right over there .	yes it 's a furnished eatin kitchen . the furniture and the icebox are all pretty outdated .
where do they go ?	they 're on the bottom floor .	i have no idea .	they re in the cabinet next to the library .	go to them department please .
let me pay for it .	ok . here you are .	cash credit card or debit card ?	ok . here is the money .	cash credit card or debit card ?
he never does and in fact he seldom communicates with him .	he is a hard job . he can also learn a lot from the government .	that 's right . he has no children and his wife experience too .	he is a very kind of man .	yeah . he 's a very easy teacher .
so you must have had a busy and enjoyable day yesterday .	yes i am . i really enjoyed it .	yes . i am very sorry i can t make it .	i will . thanks for coming .	yes . let 's go .
yes . the reservation under mr . watson .	ok .	could you spell that please ?	i 'm afraid there 's no seat in the middle of the table .	ok . would you please fill out the table ?
there are so many to choose from . i did n't know you could do all of this using a card .	i see .	what 's the best part ?	i 'm not sure . i 'll go to the doctor and get my card right away .	you mean you do n't have to tell me the truth ?
it was great ! how was your year ?	it was great . i had a great time .	my mother had a birthday party for me .	it was great . i 'm glad i did n't like it .	it was ok . there was a lot of turbulence though . the worst thing was n't too good !
tomorrow ? no i am afraid i won t be able to . i m going on business to birmingham .	i think you are looking for a .<unk><unk>in the future . the best reason why are you want to see the results .	well there is a problem with the exchange rate . it s the same size that you have in the u . s .	you should go to the dentist .	you might want to change your plans . with the cold and wet weather that we re supposed to have tomorrow the roads are going to be rather slippery .

Table 11: Responses to randomly selected test inputs which we made sure were not in the training data. *Unfiltered* is the model trained on unfiltered data, and IDENTITY TARGET is the model trained on IDENTITY, TARGET filtered data. *Overfitted* means that the respective model is evaluated at an overfitted point.

Input	**Unfiltered**	**Unfiltered-Overfitted**	**IDENTITY TARGET filtered**	**IDENTITY TARGET filtered-overfitted**
you 're flattering me .	no .	no really .	no but you can t .	no really . i can only hear a little accent .
three bags and a suitcase . this is my luggage to check .	ok . i will take one .	what 's this ? essential oil ? wow ! this is a small bag .	i 'll put you in the bag .	ok . is this your luggage properly labelled sir ?
i like orange better .	what 's your favorite flavor ?	how about drinks ?	i 'll tell you what you 're doing .	could i have some water ?
i like to watch barbie sticking her tongue out .	what 's her figure ?	oh she 's a pretty woman .	you are not supposed to be serious .	oh shut up ! she 's not what you are thinking about .
if you give us your inquiry i shall go very carefully into the price and try my best to put you on the best of the terms .	ok . i will take it .	thank you very much . we will try our best to do our best to help you .	i understand . i understand that you are right .	ok i will . i look forward to it .
ok . any time on friday will be ok with me .	ok . see you then .	i think so .	i 'll be expecting you then .	great !
no wonder you can control your voice so well . you are a professional singer .	i am not a singer .	i would n't mind a position that would be great .	i am a singer . i think i can be a singer in the lead .	i would n't like to be a singer but i love to sing that dream .
when can i get high speed internet installed ?	you can make a list of the company and make sure that you do n't have to use the machine .	you 'll need to make an appointment before the company starts .	you can get up at the airport and you 'll have to make an appointment .	you 'll have to make an appointment before we can come in and install it .
i like those kinds of programmes too . they re very informative . i think that many people underrate the education value of tv .	i agree . people often criticize tv for showing too much sex and violence .	i agree . people often criticize tv for showing too much sex and violence .	i think that teenagers should be often become addicted and violence .	i agree . people often criticize tv for showing too much sex and violence .
can you tell that i m excited ?	i m afraid i m not a child .	i m a fashion designer . next month i have a fashion show .	i m not sure . what do you want to know ?	i m sure that i ll cook the same for you .
would you like to have a shampoo sir ?	yes i 'd like a glass of iced tea .	no thank you .	no i 'd rather have it with me .	yes with the lux dandruff shampoo . it helps a lot to get rid of my dandruff .
what else would i prepare sir ?	you can bring your beef with your chicken and a bottle of wine .	oh good ! can you make some recommendation for me ?	well the interviewers are not interested in the position .	let me see . everything looks fine .
he always says i am a hard worker with consciousness of responsibility sufficient education and enough experience .	i think that maybe you are right .	what s the matter with you ?	how many years of experience will you earn ?	would you like to work in the finance department ?
what made you think that ?	i like it a lot .	i love that shirt on you .	i think it 's great . i think i ve learned a lot from different countries .	i will care for it .
i can tell you what bus to catch but you have to walk a little bit .	i do n't know .	tell me how to get a ticket and a student bus station .	i 'm sorry but i do n't have to wait .	you 're going to have a car .

Table 12: Responses to randomly selected test inputs which we made sure were not in the training data. *Unfiltered* is the model trained on unfiltered data, and IDENTITY TARGET is the model trained on IDENTITY, TARGET filtered data. *Overfitted* means that the respective model is evaluated at an overfitted point.