Improving Neural Conversational Models with Entropy-Based Data Filtering

Richard Csaky
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

Gabor 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 17 evaluation metrics to show that training on datasets filtered 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. We conduct several experiments to uncover how some of the current open-domain dialog evaluation methods behave with respect to overfitting and random data. Our software for filtering dialog data and automatic evaluation using 17 metrics is released on GitHub under an MIT license.

2 Background
Most open-domain NCMs are 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

1 github.com/ricsinaruto/NeuralChatbots-DataFiltering
2 github.com/ricsinaruto/dialog-eval
3 https://bit.ly/2Myu060
4 https://bit.ly/2T1J7MK
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., 2017c), 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; Moghe et al., 2018). 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; Gao et al., 2019), 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., 2017b; 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) and data distillation methods in general are used both in machine translation and dialog systems (Axelrod et al., 2011; Li et al., 2017a). 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 Csaky (2019) 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.

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., 2017b; 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 works resort to human evaluations (Krause et al., 2017; Fang et al., 2018), others use automatic metrics only (Olabiyi et al., 2018; Xing and Fernández, 2018; Kandasamy et al., 2017; Shalyminov et al., 2018; Xu et al., 2018b), and some use both (Shen et al., 2018a; Xu et al., 2018a; Baheti et al., 2018; Ram et al., 2018). While extensive human evaluation of the methods presented here 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 problem 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 a single target only. 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 response distribution of an utterance $s$ is a natural measure of the amount of “confusion” introduced by $s$. For example, the context $\text{What did you do today?}$ has high entropy, since it is paired with many different responses in the data, but $\text{What color is the sky?}$ 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. $\text{I don’t know}$ 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 $\text{IDENTITY}$ to the following entropy computation method. For each source utterance $s$ in the dataset we calculate the entropy of the conditional distribution $T|S = s$, i.e. given a dataset $D$ of source-target pairs, we define the target entropy of $s$ as

$$H_{tgt}(s, D) = - \sum_{(s,t) \in D} p(t|s) \log_2 p(t|s) \tag{1}$$

Similarly, source entropy of a target utterance is

$$H_{src}(t, D) = - \sum_{(s,t) \in D} p(s|t) \log_2 p(s|t) \tag{2}$$

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 $\text{What did you do today?}$, or diverse only on the surface, such as in the case of a question like $\text{How old are you?}$ (since answers to the latter are semantically close). Measuring the entropy of semantic clusters as opposed to individual utterances may improve our method by reducing data sparsity. For example $\text{How are you?}$ can appear in many forms, like $\text{How are you <name>?}$ (see Section 4.2). While the individual 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 in the case of $\text{IDENTITY}$, 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 the $\text{AVG-EMBEDDING}$ setup the representation $R(U)$ of utterance $U$ is computed by taking the average word embedding weighted by the smooth inverse frequency

$$R(U) = \frac{1}{|U|} \sum_{w \in U} E(w) \cdot 0.001 \cdot \min(1, \frac{1}{p(w)})$$

of words (Arora et al., 2017), where $E(w)$ and $p(w)$ are the embedding and the probability of word $w$ respectively. We also experiment with $\text{SENT2VEC}$, 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

$$H_{tgt}(c_s, C) = - \sum_{c_i \in C} p(c_i|c_s) \log_2 p(c_i|c_s) \tag{3}$$

where $C$ is the set of all clusters and $p(c_i|c_s)$ is the conditional probability of observing an utterance from cluster $i$ after an utterance from cluster $s$. In the context of these methods, the entropy of an utterance will mean the entropy of its cluster. Note that $\text{IDENTITY}$ is a special case of this cluster-based entropy computation method, since in $\text{IDENTITY}$ 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.

Entropy values obtained with each of these methods were used to filter dialog data in three ways. The $\text{SOURCE}$ approach filters utterance pairs in which the source utterance has high entropy, $\text{TARGET}$ filters those with a high entropy target, and finally the $\text{BOTH}$ strategy filters all utterance pairs that are filtered by either $\text{SOURCE}$ or $\text{TARGET}$. Some additional techniques did not yield meaningful improvement and were excluded from further evaluation. Clustering based on the Jaccard similarity of the bag of words of utterances only added noise to $\text{IDENTITY}$ and resulted in much worse clusters than $\text{SENT2VEC}$. Clustering single occurrences of each unique utterance (as opposed to datasets with multiplicity) lead to less useful

---

3 Based on the observed relative frequency in the data.

6 https://github.com/epfml/sent2vec
clusters than when clustering the whole dataset, probably because it resulted in less weight being given to the frequent utterances that we want to filter out. K-means proved inferior to the Mean Shift algorithm, which is a density-based clustering algorithm and seems to work better for clustering vectors of sentences. Filtering stop words before clustering did not improve the quality of clusters, probably because many utterances that we want to filter out contain a large number of stop words.

4 Data Analysis

4.1 Dataset

With 90,000 utterances in 13,000 dialogs, DailyDialog (Li et al., 2017c), our primary dataset, is comparable in size with the Cornell Movie-Dials Corpus (Danescu-Niculescu-Mizil and Lee, 2011), but contains real-world conversations. 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 (e.g. they appear twice, with different targets), remaining values rise sharply to 7. This distribution is 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.

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\(^7\) 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, utterances labeled with the highest entropy are indeed those generic sources and replies which we hoped to eliminate. See Appendix A.1 for a selection of high entropy utterances and clusters.

\(^7\)Bandwidth is like a radius in the latent space of utterance representations (Fukunaga and Hostetler, 1975).
5 Experiments

In this section the model and parameter setups are presented along with 17 evaluation metrics. Limitations of these metrics are discussed and a comparison between our filtering methods is presented on DailyDialog (Section 5.3), and other datasets (Section 5.4).

5.1 Model and Parameters

| Dataset | Type | Th. | SOURCE | TARGET | BOTH |
|---------|------|-----|--------|--------|------|
| DailyDialog | AE | 3.5 | 5.39% | 7.06% | 12.0% |
| | SC | 3.5 | 6.53% | 8.45% | 14.3% |
| Cornell | ID | 4 | - | 7.39% | 14.1% |
| Twitter | ID | 0.5 | - | 1.82% | 9.96% |

Table 1: Entropy threshold (Th.) and amount of data filtered for all datasets in the 3 filtering scenarios. ID stands for IDENTITY, AE stands for AVG-EMBEDDING, and SC for SENT2VEC.

We use transformer (Vaswani et al., 2017) as our dialog model, an encoder-decoder architecture relying solely on attention mechanisms (Bahdanau et al., 2015). 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 (see Appendix A.2 for a report of hyperparameters).

The vocabulary for DailyDialog was limited to the most frequent 16 384 words, and train / validation / test splits contained 71 517 / 9 027 / 9 318 examples, respectively.

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 17 metrics that have been applied to dialog models over the past years, briefly described below. These metrics assess different aspects of response quality, thus models should be compared on the whole set of metrics.

Response length. Widely used as a simple engagement indicator (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 (Serban et al., 2017b). Probabilities are calculated based on frequencies observed in the training data. We introduce the bigram version of this metric, to measure diversity at the bigram level as well. Utterance entropy is the product of \( H_w \) and \(|U|\), also reported at the bigram level.

https://github.com/tensorflow/tensor2tensor

Clustering and Filtering. For AVG-EMBEDDING fastText\(^9\) 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. Entropy thresholds and amount of data filtered can be found in Table 1. Generally we set the threshold so that filtered data amount is similar to the DailyDialog IDENTITY scenario. 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 various beam sizes (Graves, 2012), but greedy decoding performed better according to all metrics, also observed previously (Asghar et al., 2017; Shao et al., 2017; Tandon et al., 2017).

\(^9\)https://fasttext.cc/
**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 (Liu et al., 2016; Serban et al., 2017b; Zhang et al., 2018c). average measures the cosine similarity between the averages of word vectors of response and target utterances. 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; Shen et al., 2018a; Xu et al., 2018b), 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 (see Appendix A.3 for details). It is therefore important to only compare distinct metrics computed on the same test data.

**Bleu.** Measuring n-gram overlap between response and target is widely used in the machine learning and dialog literature (Shen et al., 2018a; Xu et al., 2018b). We report BLEU-1, BLUE-2, BLEU-3, and BLEU-4 computed with the 4th smoothing algorithm described in Chen and Cherry (2014).

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 our metrics progress during training. 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). Most metrics show a similar trend of increasing until 100k steps, and then stagnating (see Appendix A.3 for more figures).

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 due to the many identical inputs in train and test splits, because of the nature of dialog data. Most interesting are embedding metrics and BLEU scores (Section 5.3), 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 (Csaky, 2019; 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 trainings on the DailyDialog dataset both at the validation loss minimum and at an overfitted point (150 epochs).

### 5.3 DailyDialog 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. TRF, the baseline transformer model trained on unfiltered data is compared to the 9 trainings on filtered data. In all tables the 17 metrics from left to right are: response length, distinct-1, and finally, coherence, embedding average, extrema and greedy, and KL divergence. Table 3 shows the results separately for each entropy computing method are in italic (and those within a 95% confidence interval).

#### Table 2: Metrics computed at the minimum of the validation loss on the unfiltered test set (DailyDialog).

| $|U|$ | $H_w^a$ | $H_w^b$ | $H_a^a$ | $H_a^b$ | $D_k^A$ | $D_k^B$ | AVG | EXT | GRE | COH | d1 | d2 | b1 | b2 | b3 | b4 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| TRF | 8.6 | 7.30 | 12.2 | 63.6 | 93 | .330 | .85 | .540 | .497 | .552 | .538 | .0290 | .194 | .142 | .135 | .130 | .119 |
| ID | B | 9.8 | 7.44 | 12.3 | 71.9 | 105 | .315 | .77 | .559 | .506 | .555 | .572 | .0247 | .138 | .157 | .151 | .147 | .136 |
| | T | 10.9 | 7.67 | 12.7 | 83.2 | 121 | .286 | .72 | .570 | .507 | .554 | .584 | .0266 | .150 | .161 | .159 | .156 | .146 |
| | S | 9.4 | 7.19 | 11.9 | 66.4 | 98 | .462 | 1.08 | .540 | .495 | .553 | .538 | .0262 | .130 | .139 | .133 | .128 | .117 |
| AE | B | 7.9 | 7.25 | 12.0 | 57.7 | 83 | .447 | 1.05 | .524 | .486 | .548 | .524 | .0283 | .132 | .128 | .121 | .115 | .105 |
| | T | 8.6 | 7.26 | 12.1 | 61.4 | 90 | .425 | 1.12 | .526 | .492 | .548 | .529 | .0236 | .115 | .133 | .127 | .121 | .111 |
| | S | 9.0 | 7.21 | 11.9 | 65.1 | 95 | .496 | 1.16 | .536 | .490 | .548 | .538 | .0232 | .109 | .132 | .127 | .121 | .116 |
| SC | B | 10.0 | 7.40 | 12.3 | 72.6 | 108 | .383 | .97 | .544 | .497 | .549 | .550 | .0257 | .131 | .145 | .142 | .138 | .128 |
| | T | 11.2 | 7.49 | 12.4 | 82.2 | 122 | .391 | .97 | .565 | .500 | .552 | .572 | .0250 | .132 | .153 | .153 | .152 | .142 |
| | S | 11.1 | 7.15 | 11.9 | 74.4 | 114 | .534 | 1.27 | .546 | .501 | .560 | .544 | .0213 | .102 | .144 | .139 | .135 | .125 |

Table 3: Metrics computed on the unfiltered test set (DailyDialog) after 150 epochs of training. TRF refers to transformer, ID to IDENTITY, AE to AVG-EMBEDDING, and SC to SENT2VEC. SOURCE-side, TARGET-side, and filtering BOTH sides are denoted by initials. Best results are highlighted with bold and best results separately for each entropy computing method are in italic (and those within a 95% confidence interval).
### Table 4: Example inputs and responses from DailyDialog

| 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 4: Example inputs and responses from DailyDialog. BASE is trained on unfiltered data, and TARGET is the model trained on IDENTITY, TARGET filtered data. Models marked with O are evaluated at an overfitted point.

Table 2) 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 we provide an automatic, unsupervised method of detecting them based on entropy.

Every value is higher after 150 epochs of training than at the validation loss minimum (Table 3). 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 some cases the best performing model gets quite close to the ground truth performance. On metrics that evaluate utterances without context (i.e. entropy, divergence, distinct), randomly selected responses achieve similar values as the ground truth, which is expected. However, on embedding metrics, coherence, and BLEU, random responses 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. Conducting a good clustering of sentence vectors is a hard task. 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 is less harmful than the noise in clusters which we consider for filtering. Finally, Table 4 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 Appendix A.3.

### 5.4 Cornell and Twitter Results

To further solidify our claims we tested the two best performing variants of IDENTITY (BOTH and TARGET) on the Cornell Movie-Dialogs Corpus and on a subset of 220k examples from the Twit-
ter corpus. Entropy thresholds were selected to be similar to the DailyDialog experiments (Table 1). Evaluation results at the validation loss minimum on the Cornell corpus and the Twitter dataset are presented in Table 5 and Table 6, respectively. On these noisier datasets our simple IDENTITY method still managed to improve over the baseline, but the impact is not as pronounced as on the Cornell datasets. Evaluation results at the validation loss minimum on the Cornell corpus and the Twitter dataset are presented in Table 5 and Table 6, respectively. On these noisier datasets our simple IDENTITY method still managed to improve over the baseline, but the impact is not as pronounced as on the Cornell datasets. On these noisier datasets the clustering methods might work better, this is left for future work. Compared to DailyDialog there are some important distinctions that also underline that these datasets are of lesser quality. The COHERENCE metric is worse on the ground truth responses than on model responses (Table 5), and some embedding metrics and BLEU scores are better on randomly selected responses than on model responses (Table 6).

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 training of open-domain dialog models. We compared various setups in an extensive quantitative evaluation, and showed that the best approach is measuring the entropy of individual utterances and filtering pairs based on the entropy of target, but not source utterances. Some limitations of current automatic metrics and the loss function have also been shown, by examining their behavior on random data and with overfitting.

In the future, we plan to explore several additional ideas. 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 wish to conduct clustering experiments on noisier datasets and try other sentence representations (Devlin et al., 2018). We also plan to combine our method with coherence-based filtering (Xu et al., 2018b). Furthermore, we intend to perform a direct quantitative evaluation of our method based on 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.

Acknowledgments

We wish to thank Evelin Ács, Péter Ihász, Márton Makrai, Luca Szegletes, and all anonymous reviewers for their thoughtful feedback. Work partially supported by Project FIEK 16-1-2016-0007, financed by the FIEK_16 funding scheme of the Hungarian National Research, Development and Innovation Office (NKFIH).

| ID  | TRF  | B   | T   | RT  | GT  |
|-----|------|-----|-----|-----|-----|
| [U] | H^n  | H^w | H^s | H^t | D^n | D^w | D^s | D^t | AVG | EXT | GRE | COH | d1 | d2 | b1 | b2 | b3 | b4 |
| TRF | 8.1  | 6.35 | 10.4 | 54  | 75  | 2.29 | 3.40 | 0.667  | 0.451 | 0.635 | 0.671 | 4.7e-4 | 1.0e-3 | 0.108 | 0.120 | 0.120 | 0.112 |
| B   | 7.4  | 6.67 | 10.8 | 50  | 69  | 1.96 | 2.91 | 0.627  | 0.455 | 0.633 | 0.637 | 2.1e-3 | 7.7e-3 | 0.106 | 0.113 | 0.111 | 0.103 |
| T   | 12.0 | 6.44 | 10.4 | 74  | 106 | 2.53 | 3.79 | 0.646  | 0.456 | 0.637 | 0.651 | 9.8e-4 | 3.2e-3 | 0.108 | 0.123 | 0.125 | 0.118 |
| RT  | 13.4 | 8.26 | 14.2 | 113 | 170 | 0.03 | 0.12 | 0.623  | 0.386 | 0.601 | 0.622 | 4.6e-2 | 3.2e-1 | 0.079 | 0.102 | 0.109 | 0.105 |
| GT  | 13.1 | 8.18 | 13.8 | 149 | 144 | 0   | 0   | 0.655  | 0.402 | 0.567 | 0.567 | 2.0e-2 | 3.1e-1 | 1   | 1   | 1   | 1   |

Table 5: Metrics on the unfiltered test set (Cornell) at the validation loss minimum. TRF refers to transformer, ID to IDENTITY. TARGET-side, and filtering BOTH sides are denoted by initials. Best results are highlighted with bold. GT refers to ground truth responses and RT refers to randomly selected responses from the training set.

| ID  | TRF  | B   | T   | RT  | GT  |
|-----|------|-----|-----|-----|-----|
| [U] | H^n  | H^w | H^s | H^t | D^n | D^w | D^s | D^t | AVG | EXT | GRE | COH | d1 | d2 | b1 | b2 | b3 | b4 |
| TRF | 20.6 | 6.89 | 11.4 | 121 | 177 | 2.28 | 3.40 | 0.643  | 0.395 | 0.591 | 0.659 | 2.1e-3 | 6.2e-3 | 0.0519 | 0.0666 | 0.0715 | 0.0693 |
| B   | 20.3 | 6.95 | 11.4 | 119 | 171 | 2.36 | 3.41 | 0.657  | 0.394 | 0.595 | 0.673 | 1.2e-3 | 3.4e-3 | 0.0563 | 0.0736 | 0.0795 | 0.0774 |
| T   | 29.0 | 6.48 | 10.7 | 157 | 226 | 2.68 | 3.69 | 0.644  | 0.403 | 0.602 | 0.660 | 1.4e-3 | 4.6e-3 | 0.0550 | 0.0740 | 0.0819 | 0.0810 |
| RT  | 14.0 | 9.81 | 15.9 | 136 | 171 | 0.15 | 0.19 | 0.681  | 0.334 | 0.543 | 0.695 | 8.5e-2 | 5.4e-1 | 0.0444 | 0.0751 | 0.0852 | 0.0840 |
| GT  | 14.0 | 9.78 | 15.8 | 135 | 167 | 0   | 0   | 0.734  | 0.812 | 0.537 | 0.654 | 5.3e-1 | 5.3e-1 | 1   | 1   | 1   | 1   |

Table 6: Metrics on the unfiltered test set (Twitter) at the validation loss minimum. TRF refers to transformer, ID to IDENTITY. TARGET-side, and filtering BOTH sides are denoted by initials. Best results are highlighted with bold. GT refers to ground truth responses and RT refers to randomly selected responses from the training set.

https://github.com/Marsan-Ma/chat_corpus/
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.

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.

Xiang Gao, Sungjin Lee, Yizhe Zhang, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. 2019. Jointly optimizing diversity and relevance in neural response generation. arXiv preprint arXiv:1902.11205.

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, and Dan Jurafsky. 2017a. Data distillation for controlling specificity in dialogue generation. arXiv preprint arXiv:1702.06703.

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. 2017b. 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. 2017c. 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 tuning 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.

Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M. Khapra. 2018. Towards exploiting background knowledge for building conversation systems. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2322–2332, Brussels, Belgium. Association for Computational Linguistics.

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 Intelligence.
Iulian V Serban, Chinnadhurai Sankar, Mathieu Germain, Joao Sedoc, Daphne Ippolito, Arun Kirubarajan, Jai Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu 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, Zouhan Lin, Sandeep Subramanian, Tae-Sup 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 AAI, 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 Klawok. 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 Dembberg. 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, Zhi Feng Gao, Su Ke 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, Xiajun 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).
A Appendix

A.1 High Entropy Utterances

A.1.1 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 7: Top 20 source utterances (from DailyDialog) sorted by entropy. The entropy was calculated with \texttt{IDENTITY}.

A.1.2 High Entropy Clusters

Figure 6: A high entropy cluster from DailyDialog.

Figure 7: A high entropy cluster from DailyDialog.

Figure 8: A high entropy cluster from DailyDialog.
A.2 Model Parameters

| 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 8: Transformer hyperparameters.

A.3 Evaluation Metrics and Examples

Figure 9: Distinct-1 metric with respect to number of test examples (on DailyDialog). 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 (on DailyDialog). 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 (DailyDialog).

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 (DailyDialog).
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 (DailyDialog).

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 (DailyDialog).
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 (DailyDialog).
Table 9: Responses to randomly selected test inputs which we made sure were not in the training data (DailyDialog). 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.
you’re flattering me.

three bags and a suitcase. this is my luggage to check.

i like orange better.

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. any time on friday will be ok with me.

no wonder you can control your voice so well. you are a professional singer.

when can i get high speed internet installed?

i like those kinds of programmes too. they’re very informative. i think that many people underrate the education value of tv.

can you tell that i’m excited?

would you like to have a shampoo sir?

what else would i prepare sir?

he always says i am a hard worker with consciousness of responsibility sufficient education and enough experience.

what made you think that?

can you tell what bus to catch but you have to walk a little bit.

Table 10: Responses to randomly selected test inputs which we made sure were not in the training data (DailyDialog). 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.