Abstract

We propose a novel dialogue modeling framework which uses binary hashcodes as compressed text representations, allowing for efficient similarity search, and a novel lower bound on mutual information between the hashcodes of the two dialog agents, which serves as a model-selection criterion for optimizing those representations towards better alignment between the dialog participants and higher predictability of one response from another, facilitating better dialogue generation. Empirical evaluation on several datasets, from depression therapy sessions to Larry King TV show interviews and Twitter data, demonstrate that our hashing-based approach is competitive with state-of-the-art neural network based dialogue generation systems, often significantly outperforming them in terms of response quality and computational efficiency, especially on relatively small datasets.

1 Introduction

Dialogue modeling and generation is an area of active research, and of great practical importance, as it provides a basis for building successful conversational agents in a wide range of applications. While an open-domain dialogue remains a highly challenging open problem, developing dialogue systems for particular applications can be more tractable, due to specific properties of the application domain.
patient followed by a considerably shorter response of a therapist; this property is also shared with some other types of dialogues, e.g., TV show interviews such as Larry King dataset analyzed in this paper, where the guest of a show produces relatively long monologues, with the host inserting relatively short comments. Moreover, therapist’s responses tend to be rather high-level, generic statements, typically summarizing/confirming the patient’s response, and can be viewed as a “category label” being “assigned” to a patient’s text “sample”. Generating such a short, constrained response can be a somewhat easier task than solving the open-domain dialogue challenge.

Rather than generating a specific response sentence directly, a more effective approach can be to first infer a conceptual, high-level representation reflecting the essence of the “most relevant” response to the patient. Furthermore, it is desirable to optimize representation models with the objective of increasing relevance, or coherence, between the consecutive responses, as well as predictability of one response from another, to facilitate better dialog generation – the objective likely to be captured by the notion of mutual information between the responses.

Note that a fundamental concept in psychotherapy is the working alliance between the therapist and the patient or, more generally, the client seeking help [Bordin, 1979]. The alliance involves several cognitive and emotional components of the relationship between these two agents, including the agreement on the goals to be achieved and the tasks to be carried out, and the bond, trust and respect to be established over the course of the therapy. While an encompassing formalization of working alliance is a challenging task, it is however reasonable to consider that the mutual information criterion may capture, to some extent, the dynamics of agreement expected to develop in most therapies, and, more generally, in other types of dialogues.

Motivated by above considerations, we introduce here a novel dialogue modeling framework where responses are represented as locality-sensitive binary hash-codes [Kulis and Grauman, 2009] Joly and Buisson, 2011, Garg et al., 2018], and the hashing models are optimized using a novel mutual-information lower bound, as mutual information computation is intractable in high-dimensional spaces. Using hashcode representations may allow for a more tractable way of predicting responses in a compressed, general representation space instead of direct generation of textual responses. (Note that hashcode representations were successfully applied in prior work on information-extraction [Garg et al., 2018]). Furthermore, separating inference in representation space from text generation (based on inferred response representation) increases method’s flexibility, while mutual information criterion facilitates better alignment between the responses of two parties and higher predictability of the proper response. While the psychotherapy domain was our primary motivation, the proposed approach is generally applicable to a wider range of domains as demonstrated in empirical section. Overall, our key contributions include: (1) a novel generic framework for dialogue modeling and generation using locality sensitive hash functions; (2) a novel lower bound on the Mutual Information (MI) between the hashcodes of the responses from the two agents used as an optimization criterion for the locality sensitive hash functions; (3) an extensive empirical evaluation on three different dialogue domains, from depression therapy to TV show interviews and Twitter data, demonstrating competitiveness of our approach when compared with the state-of-art neural network based dialog systems, and its clear superiority in case of relatively small datasets (e.g., therapy sessions), both in terms of the quality of generated responses and computational efficiency.

2 Related Work

Therapy chatbots, such as, for example, Woebot [Fitzpatrick et al., 2017] and similar recently developed dialog systems, often based on Cognitive Behavioral Therapy (CBT) [Lewinsohn et al., 1990], are becoming increasingly popular; however, these agents have limited ability to understand free text and have to resort to a fixed set of preprogrammed responses to choose from [Di Prospero et al., 2017, Ly et al., 2017, Schroeder et al., 2018, Morris et al., 2018, Hamamura et al., 2018]. (Also, see Jurafsky and Martin, 2014 for an overview.)

For dialogue modeling in general domains, several recently proposed neural network based approaches are considered state-of-art [Serban et al., 2015, Serban et al., 2016a, Serban et al., 2017a, Serban et al., 2017b, Shao et al., 2017, Asghar et al., 2017, Wu et al., 2017]. However, those approaches are usually data-hungry and may not perform well on relatively small datasets, such as transcribed therapy sessions (as we observe in this paper); furthermore, they are not typically explored in dialogue settings where one person’s response can be extremely long, e.g. up to tens of thousands of words, as in case of patient’s responses in therapy datasets considered here. Also, evaluating the effectiveness of the therapist’s response requires some notion of relevance which goes beyond the standard measures of its semantic features [Papineni et al., 2002, Liu et al., 2016, Li and Jurafsky, 2016, Lowe et al., 2017, Li et al., 2017]; we consider here an information-theoretic approach to capture this notion.

Unlike task-driven dialogue [Zhai and Williams, 2014, Wen et al., 2016, Althoff et al., 2016, Lewis et al., 2017, He et al., 2017], an immediate response quality metric may not be available in our settings, since the effect of therapy is harder to evaluate and multiple sessions are often
required to achieve the desired outcome; thus, some intermediate criteria must be introduced. Attention to specific parts of the response, as well as background knowledge, explored in neural network-based dialogue modeling [Kosovan et al., 2017] can be also helpful in therapeutic dialogues; those aspects are, to some extent, implicitly captured by learning the hashing models. Note that in related work by [Bartl and Spanakis, 2017], hashcodes are not treated as representations of the responses, and are only used for the nearest neighbor search, unlike the approach proposed here.

While mutual information has been previously considered in dialogue modeling [Li et al., 2015], it was only applied during testing, unlike our work which uses it as a model selection criterion for learning representations in training stage.

Note that popular metric such as BLEU score [Papineni et al., 2002] does not focus on relevance between the two responses, but rather tries to capture all information when comparing the ground truth with the produced text.

Regarding the mutual information estimation from data, there are many existing approaches based on k-Nearest Neighbors, Random Forest, ensemble models, etc [Barber and Agakov, 2003, Kraskov et al., 2004, Koeman and Heskes, 2014, Singh and Poczos, 2014, Gao et al., 2015] [Moon et al., 2017]. However, in high dimensional settings, these estimators are highly expensive, and quite inaccurate when the number of samples is relatively small. Previously, several mutual information lower bounds have been proposed for classification problems [Chalk et al., 2016, Gao et al., 2016, Alemi et al., 2017], assuming one-dimensional class label; unfortunately, they do not apply in our setting where the predicted response is a vector in some high-dimensional representation space.

Figure 1: An intuitive illustration: the objective behind learning a hashing representation via maximizing the mutual information between a patient (left) and therapist (right) responses is to find a compressed encoding of those responses which preserves the mutually relevant content while ignoring irrelevant details; e.g., in the example above, we would expect a good representation model to capture the content highlighted in boldface as essential to the conversation.

The essence of our approach is to select hashing-based representation models, from a large model space determined by multitude of hyper-parameters described later, so that the response of the person B (e.g., therapist) is maximally relevant given the response of the person A (e.g., patient), as measured by the mutual information between the two (see Figure 1). From another perspective, this objective is also aimed at learning representations making the second response maximally predictable given the first response.

3 Problem Formulation and Background

We now present a novel framework for dialogue modeling using binary hash functions. We will refer to the two dialogue agents as to a patient and a therapist, respectively, although the approach is generally applicable to a wider variety of dialogue settings, as demonstrated later in the empirical section on datasets such as TV show interviews and Twitter dialogues.

3.1 Problem Formulation and Approach Overview

We consider a dialogue dataset consisting of \( N \) samples, \( S^p = \{ S^p_1, S^p_2, \ldots, S^p_N \} \), where each sample is a pair of a patient and a therapist responses, \( S^p_i \) and \( S^t_i \), respectively; we will also use the following notation: \( \bar{S} = \{ S^p_1, \ldots, S^p_N, S^t_1, \ldots, S^t_N \} \). Each response is a natural language structure which can be simply a text, or a text with part of speech tags (PoS), or a syntactic/semantic parsing of the text.

Given response \( S^p_i \), the dialogue generation task is to produce the response \( S^t_i \). We approach this task as a three-stage problem: first, we learn a representation model, based on locality sensitive hashing, which maps each text response \( S_i \) into some binary hashcode vector \( c_i \in \{0, 1\}^H \); second, we train a classifier to infer the therapist’s hashcode \( c_i^t \) given the patient’s hashcode \( c_i^p \), so that the inference takes place in the abstract representation space; hashcode representation aims at capturing, in a compressed form, the semantic essence of the responses while leaving out irrelevant details; finally, we produce a textual response based on the predicted hashcode representation.

The essence of our approach is to select hashing-based representation models, from a large model space determined by multitude of hyper-parameters described later, so that the response of the person B (e.g., therapist) is maximally relevant given the response of the person A (e.g., patient), as measured by the mutual information between the two (see Figure 1). From another perspective, this objective is also aimed at learning representations making the second response maximally predictable given the first response.

\[ \text{Here we use a simple approach to text generation, choosing the nearest hashcode from all response hashcodes in a training dataset (sublinear time algorithms can be used for efficient similarity search in hamming spaces [Norouzi et al., 2014, Komorowski and Trzcinski, 2017]); however, a wider range of responses can be generated by using any unsupervised natural language generation model within our framework [Bowman et al., 2015, Jozefowicz et al., 2016, Semeniuta et al., 2017, Yu et al., 2017].} \]
hash functions, and let \( h \) be a function here. For instance, as per a learned locality sensitive hashing model learned for a given task (\( S \) refers to the reference set), both \( S \) and \( \bar{S} \) denote a set of \( \{S_i\} \mid i = 1, \ldots, M \). Further, let \( h_l(S_i) \) denote \( \alpha \) with low hamming distance to each other, are similar to each other, locality sensitive hashcodes should serve as generalized representations of language structures (a similarity/distance function implied as per the locality sensitive hashing model learned for a given task)\(^3\) and so for the responses in a dialogue. There are multiple hash functions proven to be locality sensitive\(^4\)\([98]\). Recently, several kernel-based locality-sensitive hashing approaches have been developed that are applicable for natural language processing\([2009, 2011, 2018]\). These techniques rely on a convolution kernel similarity function \( K(S_i, S_j; \theta) \) defined for any pair of structures \( S_i \) and \( S_j \) with kernel parameters \( \theta \)\([2013, 2005]\); see the supplementary materials for more details.

In order to construct hash functions for mapping textual responses to hashcodes, we will first select from a training dataset a random subset of text structures (responses) \( S^R \subset \bar{S} \) of size \( |S^R| = M \), called a reference set. Further, let \( h_l(S_i) \), \( l = 1, \cdots, H \), denote a set of \( H \) binary-valued hash functions, and let \( h(S_i) \) denote vector \( \{h_l(S_i)\}_{l=1}^{H} \) of size \( \alpha \ll M \).

The hashcode representation of response \( S_i \) will be given as \( c_i = h(S_i) \).

We will now describe the procedure for generating the above hash functions \( h_l(S_i) \). For each bit \( l \), we first select a random subset \( S^R_l \subset \bar{S} \) of the reference set, \( |S^R_l| = 2\alpha \). Next, we assign label 0 to \( \alpha \) randomly selected elements of \( S^R_{l} \), and label 1 to the remaining \( \alpha \) elements of that set, creating an artificial binary-labeled training dataset, which can be now fed into any binary classifier to learn a function \( h_l(S_i) \). For example, Fig. 2 illustrates this approach using the Random Maximum Margin (\( LSH-RMM \)) method\([2011]\), where maximum-margin classifier (SVM) is applied to random splits of the reference set into blue (label 0) and red (label 1) points, respectively. We generate \( H \) such random splits of the reference set, and learn the corresponding \( H \) binary classifiers, or hash functions. We also tried k-nearest neighbor classifier (kNN), resulting into hashing approach we refer to as \( LSH-RkNN \).

Overall, to obtain a hashcode of a given response \( S_i \), we must compute \( M \) kernel similarities, \( K(S_i, S_j; \theta) \), \( \forall S_j \in S^R \). For a limited size (\( M \)) of the reference set \( S^R \), hashcodes can be computed efficiently, with the computational cost linear in \( H \); also, note that LSH techniques described above are easily parallelizable\([98]\).

Since an arbitrary classifier can be used in the above generic approach\([98]\), we also experimented with a neural language model such as LSTM trained via standard back-propagation, obtaining a neural hash function, referred to as \( LSH-RLSTM \) in this paper; note that there is no need to decide on the choice of the reference set since LSTM model can easily handle large training datasets; however, there are other model-selection choices such as network’s architecture parameters that need to be optimized.

### 4 Learning Hashcode Representations

Given that each specific hashing model described above involves several model-selection choices, our task will be to optimize those choices using the information-theoretic criterion proposed below.

#### Optimizing LSH Model Parameters

As per the discussion of LSH above, an LSH model involves the function \( Lsh(\cdot; \theta, S^R) \) for mapping text responses to hashcodes:

\(1\)One can also use other kernel based classifiers like Gaussian processes\([2006]\) to build hash functions within LSH. The approach of\([2009]\) also comes under the same LSH framework. In their work, a hash function \( h_l(\cdot) \) is built as a random linear hyperplane in the kernel implied feature space, that is computed approximately using the random subset \( S^R \).

\(^3\)While theoretical guaranties for locality sensitivity of the above described hashing scheme are missing in the literature, the principle of randomizing each hash function by learning it on a very small subset of samples \( S^R \) of size \( 2\alpha \ll M \), should suffice in practice\([2014]\).
\(c_t = lsh(S_t; \theta, S^R),\) where \(c_t = h(S_t) = (h_i(S_t))_{i=1}^H,\)
and where each hash function \(h_i(.)\) is built based on a random subset of \(S^R\) using either a kernel (kNN, SVM) or a neural network (LSTM) classifier. For the case of kernel-based LSH, \(\theta\) are the parameters of a convolution kernel similarity function \(K(S_i, S_j)\). For neural hashing (LSH-RLSTM), \(\theta\) refers to the neural architecture hyperparameters (number of layers, the number of units in a layer, type of units, etc.); \(\theta\) also includes LSH-specific parameters such as \(\alpha\).

When learning LSH models on a training dataset, the (hyper) parameters \(\theta\) as well as the reference set \(S^R\) will be optimized with respect to the information-theoretic objective introduced below. Namely, for LSH-RkNN and LSH-RMM, the kernel parameters \(\theta\) are optimized via grid search. For LSH-RLSTM, \(\theta\) reflects the neural architecture, i.e., the number of layers and the number of units in each layer, optimized by greedy search. Similarly, \(S^R\) is also constructed via a greedy algorithm. Pseudo codes for the greedy algorithms are presented in the supplementary material.\(^6\)

**Info-theoretic Objective Function.** The objective function for hashcode-based model selection in dialog generation should (1) characterize the quality of the hashcodes as generalized/compressed representations of dialogue responses and (2) favor representation models leading to higher-accuracy response generation.

Mutual information \(I(S^p : S^t)\) between the dialog responses \(S^p\) (e.g., patient) and \(S^t\) (e.g., therapist) is a natural candidate objective as it implies higher predictability of one response from another. Though, it is hard to compute in practice as the joint distribution over all pairs of textual responses is not available. However, we can attempt to approximate it using hashcode representations. If \(h(.)\) represents a function from the space of all statements to the hash code space, then the data processing inequality implies that \(\forall h(.), I(S^p : S^t) \geq I(h(S^p) : h(S^t))\), and maximizing the quantity on the right can be more computationally feasible.

Thus we will maximize the mutual information (MI) between the response hashcodes, over LSH model parameters; it turns out that MI reflects both the inference accuracy as well as the representation quality, as we will see below:

\[
\arg \max_{\theta, S^R} I(C^p : C^t); \tag{1}
\]

\[
C^p = lsh(S^p; \theta, S^R), \quad C^t = lsh(S^t; \theta, S^R) \tag{2}
\]

\[
I(C^p : C^t) = H(C^t) - H(C^t|C^p) \tag{3}
\]

Herein, \(C^p\) and \(C^t\) are the multivariate binary random variables associated with the hashcodes of patient and therapist responses, respectively. Minimizing the conditional entropy, \(H(C^t|C^p)\), improves the predictive accuracy when inferring therapist response hashcode, while maximizing the entropy term, \(H(C^t)\), should ensure good quality of the hashcodes as generalized representations of text responses; thus MI objective satisfies both criteria stated at the beginning of this section.\(^6\)

### 4.1 Information-Theoretic Bounds

Since computing mutual information between two high-dimensional variables can be both computationally expensive and inaccurate if the number of samples is small [Kraskov et al., 2004, Walters-Williams and Li, 2009, Singh and Poczos, 2014, Gao et al., 2015], we develop a (novel) lower bound on the mutual information which is easy to compute; derivation details are given in the supplementary material. We will first introduce the information-theoretic quantity called Total Correlation [Watanabe, 1960], \(TC(C) = \sum_j H(C_j) - H(C)\), which captures non-linear correlation among the dimensions of a random variable \(C\); given an additional random variable \(Y\), \(TC(C : Y)\) is defined as \(TC(C : Y) = TC(C) - TC(C|Y)\).

**Theorem 1 (Lower Bound on Mutual Information).** Mutual information between two random hashcode variables, \(I(C^t : C^p),\) can be bounded from below as follows:

\[
I(C^t : C^p) \geq \sum_j H(C_j) - TC(C^t : Y^*) + \sum_j \log q(C_j^t|C^p) \tag{4}
\]

\[
Y^* \leftarrow \arg \max_{Y : |Y| = |C^p|} TC(C^t : Y) \tag{5}
\]

Herein, \(TC(C^t : Y)\) describes Total Correlations within \(C^t\) that can be explained by a latent variables representation \(Y; q(C_j^t|C^p)\) is a proposal conditional distribution \(^6\)

---

\(^6\)Since parameters under optimization are shared between all the hash functions jointly, this allows keeping individual hash functions randomized (built from small random subsets of \(S^R\)), which should possibly help in learning hashcode-based representations that are more robust to over-fitting a dialog training dataset. Further, although it is theoretically difficult to establish, this inherent randomization of hash functions, despite the learning of the shared parameters, should help towards keeping hashcodes locality-sensitive (an important property for hashcodes to be good feature representations).

\(^6\)If we were learning LSH parameters by minimizing the conditional entropy term \(H(C^t|C^p)\), only, entropy of the hashcodes \(H(C^t)\) could also reduce since the latter reduction would lead to a decrease in the intrinsic dimension of the hashcodes inference problem itself, thus making the inference more accurate, corresponding to low conditional entropy. This kind of decrease in the conditional entropy would not be desirable as the problem of inferring hashcodes itself would be somewhat artificial.
for the \( j \)th bit of the hashcode \( C^t \) predicted using a probabilistic classifier, like a Random Forest model.

As discussed in [Ver Steeg and Galstyan, 2014], \( \TC(C^t : Y^*) \) can be computed efficiently.

Note that the first two terms in the MI lower bound contribute to improving the quality of hashcodes as response representations, maximizing entropy of each hashcode bit while discouraging redundancies between the bits, while the last term containing conditional entropies aims at improving inference of individual hashcode bits.

Moreover, we will also use the proposed MI LB as an evaluation metric of the dialog quality on test data, i.e. the alignment/relevance between the responses of two dialog agents. We also use a normalized metric, dividing MI LB by an upper bound on joint entropy \( \sum_i \mathcal{H}(C_i^t) - \sum_i \mathcal{H}(C_i^t : Y^*) \leq \sum_i \mathcal{H}(C_i^t) - \mathcal{H}(Y^*) \). For \( \mathcal{H}(C_i^t | Y^*) = 0 \), i.e. when a latent representation \( Y^* \) is learned which explains all the Total Correlations in \( C_i^t \), the upper bound becomes equal to the entropy term; practically, for the case of hashcodes, learning such a representation should not be difficult, so the bound should be tight.

5 Empirical Evaluation

Several variants of the proposed hashing based dialog model, using kNN, SVM or LSTM to build hashcodes, respectively, were evaluated on three different datasets and compared with three state-of-art dialog generation approaches of [Serban et al., 2017b] and [Vinyals and Le, 2015]. Besides standard evaluation metrics adopted by those approaches, we also used several hashcode-model specific metrics.

5.1 Experimental Setup

Datasets. The three datasets used in our experiments include (1) depression therapy sessions, (2) Larry King TV interviews and (3) Twitter dataset. The depression therapy dataset consists of transcribed recordings of nearly 400 therapy sessions between multiple therapists and patients. Each patient response \( S^A_i \) followed by therapist response \( S^B_i \) is treated as a single sample; all such pairs, from all sessions, were combined into one set of \( N=42000 \) samples. We select 10% of the data randomly as a test set (4200 samples), and then perform another random 90/10 split of the remaining 38,000 samples into training and validation subsets, respectively. We follow the experimental setup from prior work cited above when comparing the respective neural network models with our hashing based approaches: all models are trained only once using the same training and validation datasets, and evaluated on the same test set. However, for our hashing model metrics introduced below, we average the estimates over 10 random subsets using 95% of test samples each time.

The Larry King dataset contains transcripts of interviews with the guests of TV talk shows, conducted by Larry King, the host. Similarly to the depression therapy dataset, we put together all pairs of guest/host responses from 69 sessions into a single set of size 8200. The data are split into training, validation and test subsets as described earlier.

Next, we experimented with the Twitter Dialogue Corpus [Ritter et al., 2010]. Considering the original tweet and the following comments on it, in the same session, the task is to infer the next tweet. Note that we consider all utterances preceding that tweet as one long utterance, i.e. as the first “response” \( S^A \), mapped to one hashcode, while the next tweet is the second “response” \( S^B \), which is different from the approach of [Serban et al., 2017b] we compare with, where the previous utterances in a session are explicitly viewed as a sequence. The number of tweet sessions (each viewed as a separate sample, i.e. \( \{S^A_i,S^B_i\} \) pair of responses), in training, validation, and test subsets are, respectively, 749060, 93633, and 93633.

Task. For all datasets, the task is to train a model on a set of training samples, i.e. response pairs \( (S^A_i, S^B_i) \), where \( S^A \) is a response of person A, followed by the corresponding response of a person B. Then each test sample is given as a response of person A, and the task is to generate the response of a person B.

Hashing Models.

Step 1: Representation Learning. We evaluate three different hashing models: the first two, based on kernel locality sensitive hashing (KLSH) [Joly and Buisson, 2011] and [Garg et al., 2018], are called \( \text{LSH-RMM} \) and \( \text{LSH-RkNN} \), and they use, respectively, Max-Margin approach of SVM classifier (with \( \beta = 1 \) parameter) or kNN classifier (\( k = 1 \)), to compute each hash function; \( R \) in both names stands for random data splits used to compute each hash function. The third hashing model, denoted \( \text{LSH-RLSTM} \), uses LSTM to compute each hash function. We use hashcode vectors of dimensionality \( H=100 \). For \( \text{LSH-RkNN} \) and \( \text{LSH-RMM} \), we use as a reference set a random subset of \( M=300 \) samples from the training dataset, to reduce the computational complexity of training those models, but for \( \text{LSH-RLSTM} \) we use the whole training dataset as a reference set. Parameters \( \theta \) for LSH models are obtained by maximizing the proposed MI LB criterion, using \( \gamma = 1000 \).
We now map all responses, Step 2: Hashcode Prediction. tails on the LSH models. reported in [Serban et al., 2017b], without rerunning the models; standard deviations were not reported in that paper. based similarity metrics between the actual and generated responses. Mean and standard deviation across samples (response pairs) are RkNN, LSH-RMM and LSH-RLSTM), on three datasets – Depression Therapy, Twitter, and Larry King data – using word embedding-based similarity metrics between the actual and generated responses. Mean and standard deviation across samples (response pairs) are Table 2: Comparison between state-of-art neural network models (LSTM, HRED and VHRED) and the proposed hashing models (LSH-RkNN, LSH-RMM and LSH-RLSTM), on three datasets – Depression Therapy, Twitter, and Larry King data – using word embedding-

See the supplementary material for more experimental details on the LSH models.

Step 2: Hashcode Prediction. We now map all responses, of both participants A and B, in both training and test sets, to the corresponding hashcodes using one of the above hashcode-based representation models. Next, to predict the response hashcode of a person A given a hashcode of a person B, we train separate Random Forest (RF) classifiers (each containing 100 decision trees) for each hashcode bit (i.e. 100 such RF classifiers, since H=100).

Step 3: Textual Response Generation. Given a hashcode of a response inferred by RF classifier above, mapping it to an actual text can be performed in multiple ways; for now, we simply find the nearest neighbor of the generated hashcode in the set of all hashcodes corresponding to the person B responses in our training data.

Baseline: Neural Network Dialog Generation Models. We compare our dialog generation method with the state-of-art VHRED approach of [Serban et al., 2017b], as well as with the two other approaches, HRED [Serban et al., 2016b], and LSTM [Vinyals and Le, 2015], also used as baselines in the VHRED paper. We adopt the same hyperparameter settings as those used in [Serban et al., 2017b]. For the Twitter dataset, we compare with the results presented in the above paper, while on the other two datasets, we train the above models ourselves. The vocabulary size for the input is set via grid search between values 1000 to 100000. The neural network structures are chosen by an informal search over a set of architectures and we set maximum gradient steps to 80, validation frequency to 500 and step-size decay for SGD is 1e-4.

Evaluation metrics. Embedding-based metrics. We compare our methods with the state-of-art neural network approaches listed above using three word embedding-based topic similarity metrics - embedding average, embedding greedy, and embedding extrema [Liu et al., 2016], adopted by [Serban et al., 2017b]. Following the prior art, we used Google News Corpus to train the embeddings[10]. Given a textual response $S^M$ (of Person B) generated by a particular method, and the true textual response $S^A$ (of person A), all words are first mapped to their corresponding embeddings. An average across the words in each response is computed, and the cosine between the two resulting vectors constitutes the embedding average similarity metric. Another approach to computing response-level embeddings is to use vector extrrema, where, for each dimension of the word vectors, we select the most extreme value amongst all word vectors in the response, and use that value in the response-level embedding; the cosine similarity is then computed between the corresponding response-level embeddings, resulting into a metric called embedding extrema [Liu et al., 2016]. Finally, the third metric, embedding greedy, does not compute response-level embeddings. Instead, given two responses $S^M$ and $S^A$, each token $w \in S^M$ is greedily matched with a token $w' \in S^A$ having maximum cosine similarity of the corresponding word embeddings, and the total score is averaged across all words $w$ [Liu et al., 2016]. Information-theoretic metrics. The second group of metrics directly evaluates the quality of hashcodes obtained using our models. For each method, we will report the proposed MI lower bound (MI LB) from Theorem 1, as well as its normalized version (NMI LB). For each of the metrics, higher values mean better performance.

Finally, we report the hashcode inference accuracy (HIA), i.e. the accuracy of predicting hashcode bits of responses B using RF classifiers. We also obtain the baseline accuracy (Baseline), using a trivial classifier that always chooses the most-frequent class label.

The mean and standard deviation statistics for each metric are computed over 10 runs of the experiment, as mentioned above; in the case of the accuracy metric, the statistics are

https://code.google.com/archive/p/word2vec/

| Model       | Depression Therapy Dataset | Twitter Dataset | Larry King Dataset |
|-------------|-----------------------------|-----------------|-------------------|
|             | Average | Greedy | Extrema | Average | Greedy | Extrema | Average | Greedy | Extrema |
| LSTM [Vinyals and Le, 2015] | 0.61 ± 0.31 | 0.58 ± 0.29 | 0.28 ± 0.16 | 0.51 | 0.39 | 0.37 | 0.71 ± 0.24 | 0.60 ± 0.20 | 0.35 ± 0.14 |
| HRED [Serban et al., 2016b] | 0.48 ± 0.23 | 0.43 ± 0.20 | 0.29 ± 0.16 | 0.50 | 0.38 | 0.36 | 0.71 ± 0.25 | 0.61 ± 0.20 | 0.29 ± 0.12 |
| VHRED [Serban et al., 2017b] | 0.53 | 0.40 | 0.38 | 0.70 ± 0.24 | 0.72 ± 0.25 | 0.43 ± 0.18 |
| LSH-RkNN    | 0.55 ± 0.39 | 0.47 ± 0.29 | 0.26 ± 0.21 | 0.61 ± 0.17 | 0.40 ± 0.13 | 0.25 ± 0.09 | 0.76 ± 0.28 | 0.60 ± 0.21 | 0.34 ± 0.15 |
| LSH-RMM     | 0.56 ± 0.38 | 0.53 ± 0.33 | 0.31 ± 0.23 | 0.61 ± 0.17 | 0.41 ± 0.13 | 0.25 ± 0.09 | 0.73 ± 0.28 | 0.59 ± 0.22 | 0.35 ± 0.16 |
| LSH-RLSTM   | 0.64 ± 0.37 | 0.51 ± 0.28 | 0.28 ± 0.19 | 0.60 ± 0.18 | 0.39 ± 0.13 | 0.24 ± 0.09 | 0.76 ± 0.27 | 0.58 ± 0.21 | 0.33 ± 0.15 |
computed over all (100) hashcode bits and over 10 trials.

5.2 Results

Embedding-based Metrics. Table 2 summarizes performance of all methods with respect to the embedding-based metrics, with the best results for each metric/column shown in boldface. Overall, the proposed hashing approaches are quite competitive with the neural net methods: on all three datasets, our LSH methods always outperform are quite competitive with the neural net methods: on all three datasets, our LSH methods always outperform...metrics, with the best results for each metric/column shown in boldface. Overall, the proposed hashing approaches are quite competitive with the neural net methods: on all three datasets, our LSH methods always outperform

Furthermore, it is important to note that the above metrics, although commonly used for evaluating dialog generating systems, do not always capture well the quality of the produced dialogs and human evaluation is still hard to replace. For example, on Depression dataset, due to its relatively small size, none of the neural net models managed to actually learn how to generate good responses; those models kept generating practically same response, of questionable quality on the majority of test samples; see several examples of actual dialogs below; on the other hand, hashing methods varied there responses, and often appeared to be quite aligned with the essence of the conversation:

See Appendix for more examples of actual dialogues; while some of the responses generated by hashing methods were making less sense than others, overall they appeared to be at least to some degree relevant to the conversation, even in relatively small datasets such as Depression and Larry King, not handled well by neural net approaches. Clearly, pretraining hashing models on larger datasets can help, even using our simple text-generation part of the approach where we select a response from the training data with the predictability of B’s responses given A’s response. You go, “Okay, well what are you avoiding?” in essence. I mean, there’s truth to that. Because I... because working in here, I agree, that’s a thing I want to do is be focused on how I’m feeling. This whole thing with my family not being able to show feelings. By the way, I just have to drive over this scenic outlook and tell you that I, you know, I did talk to my mother about... <therapist> How are you doing that now? <LSH-RkNN> It’s up to you. <LSH-RMM> Yeah, people get really provocative. <LSH-RLSTM> How so? <LSTM> I don’t want to be a lot a lot a lot a lot a lot... of the way... <HRED> M-hmm. you... <VHRED> Uh-huh. you

(b) Twitter Dataset

Table 3: Hashcode models quality as measured by the alignment between the hashcodes of person A and person B responses (mutual information lower bound and its normalized version), as well as by the predictability of B’s responses given A’s response.

| Model       | MI LB (Shuffled) | NMI LB | HIA (Baseline) |
|-------------|------------------|--------|----------------|
| LSH-RkNN    | 17.3±0.1 (9.1±0.1) | 0.75 (0.40) | 0.85±0.18 (0.81±0.21) |
| LSH-RMM     | 23.4±0.2 (11.2±0.3) | 0.71 (0.34) | 0.80±0.14 (0.72±0.16) |
| LSH-RLSTM   | 41.9±0.1 (24.3±0.1) | 0.83 (0.48) | 0.69±0.16 (0.63±0.16) |

(c) Larry King Dataset

| Model       | MI LB (Shuffled) | NMI LB | HIA (Baseline) |
|-------------|------------------|--------|----------------|
| LSH-RkNN    | 9.4±0.3 (5.1±0.3) | 0.63 (0.34) | 0.91±0.14 (0.89±0.16) |
| LSH-RMM     | 22.4±0.9 (4.5±1.2) | 0.62 (0.13) | 0.69±0.11 (0.59±0.10) |
| LSH-RLSTM   | 48.9±0.3 (28.4±0.5) | 0.80 (0.47) | 0.62±0.10 (0.54±0.07) |

| (a) Depression Therapy Dataset | Model | MI LB (Shuffled) | NMI LB | HIA (Baseline) |
|--------------------------------|-------|------------------|--------|----------------|
| LSH-RkNN                        | 12.8±0.5 (6.1±0.2) | 0.57 (0.27) | 0.87±0.16 (0.82±0.18) |
| LSH-RMM                        | 13.7±0.2 (1.0±0.3) | 0.39 (0.0) | 0.68±0.10 (0.59±0.10) |
| LSH-RLSTM                     | 20.3±0.3 (10.6±0.2) | 0.76 (0.40) | 0.82±0.19 (0.79±0.21) |

Info-theoretic Metrics. Next, we took a deep dive into evaluation of hashing approaches with respect to how well they actually model the alignment between the responses; the results are summarized in Table 3, presenting the mutual information lower bound (MI LB), normalized MI LB, and hashcode inference accuracy (HIA) as discussed be-
Moreover, in Tab. 4 we demonstrate some model improvement in terms of information-theoretic metrics when using optimized rather than randomly selected reference set in our kernel-based models; when the reference set size is small (M=100 used in our experiments), optimization can provide clear benefits over random subset selection. In the supplementary material, we show the optimized reference set for LSH-RkNN for Depression Therapy dataset.

**MI Evolution over Therapy Sessions.** Finally, we provide an initial evaluation of our (normalized) MI LB metric (using RkNN model) as a proxy of developing alliance between the patient and therapist during therapy sessions, on Depression dataset. The NMI LB metric evolution over the course of a session is computed using moving window of size 20, for 170 sessions of sufficient length (at least 100 samples, i.e. patient/therapist response pairs); next, we warp the obtained varying-length time series onto a uniform 1,000-dimensional grid using linear interpolation, so that the first and the last entries of each session are mapped to the first and last element of the grid. We then compute SVD on the resulting matrix. Fig. 3 shows the first (temporal) SVD component (i.e. the mean across the sessions), which accounts for 30% of the variance. Note an interesting temporal evolution of the metric, which can potentially be used as a proxy for evaluating alliance development (or the lack of it) during therapy sessions, as discussed earlier.

**6 Conclusions**

This paper introduces a novel approach to dialogue modeling based on hash functions, using psychotherapy sessions as a motivating domain. In our framework, responses from both parties (e.g., patient and therapist) are represented by the corresponding hashcodes, capturing certain text patterns. Furthermore, we propose a novel lower bound on Mutual Information in order to characterize the relevance of a therapist’s response to the patient’s text, and vice versa. We performed empirical evaluation of the proposed approach on several datasets, including depression therapy data, as well as more generic dialogues such as TV show interviews and Twitter data. We optimized locality sensitive hashing models, based on kernel functions or neural language models, by maximizing the proposed MI bounded objective function. Our results consistently demonstrate superior performance of the proposed approach over state-of-art neural network dialogue models, especially on relatively small datasets which are presenting a particular challenge to the neural models but are successfully handled by our approach, both in terms of training speed and response quality.

**References**

[Alemi et al., 2017] Alemi, A., Fischer, I., Dillon, J., and Murphy, K. (2017). Deep variational information bottleneck.

[Althoff et al., 2016] Althoff, T., Clark, K., and Leskovec, J. (2016). Large-scale analysis of counseling conversations: An application of natural language processing to
mental health. Transactions of the Association for Computational Linguistics, 4:463.

[Asghar et al., 2017] Asghar, N., Poupart, P., Jiang, X., and Li, H. (2017). Deep active learning for dialogue generation. In Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017), pages 78–83.

[Barber and Agakov, 2003] Barber, D. and Agakov, F. (2003). The im algorithm: a variational approach to information maximization. In Proceedings of the 16th International Conference on Neural Information Processing Systems, pages 201–208. MIT Press.

[Bartl and Spanakis, 2017] Bartl, A. and Spanakis, G. (2017). A retrieval-based dialogue system utilizing utterance and context embeddings. arXiv preprint arXiv:1710.05780.

[Bordin, 1979] Bordin, E. S. (1979). The generalizability of the psychoanalytic concept of the working alliance. Psychotherapy: Theory, research & practice, 16(3):252.

[Bowman et al., 2015] Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. (2015). Generating sentences from a continuous space. arXiv preprint arXiv:1511.06349.

[Chalk et al., 2016] Chalk, M., Marre, O., and Tkacik, G. (2016). Relevant sparse codes with variational information bottleneck. In Advances in Neural Information Processing Systems, pages 2172–2180.

[Di Prospero et al., 2017] Di Prospero, A., Norouzi, N., Fokaefs, M., and Litoiu, M. (2017). Chatbots as assistants: an architectural framework. In Proceedings of the 27th Annual International Conference on Computer Science and Software Engineering, pages 76–86. IBM Corp.

[Fitzpatrick et al., 2017] Fitzpatrick, K. K., Darcy, A., and Vierhile, M. (2017). Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (woebot): A randomized controlled trial. JMIR Mental Health, 4(2):e19.

[Gao et al., 2015] Gao, S., Ver Steeg, G., and Galstyan, A. (2015). Efficient estimation of mutual information for strongly dependent variables. In Artificial Intelligence and Statistics, pages 277–286.

[Gao et al., 2016] Gao, S., Ver Steeg, G., and Galstyan, A. (2016). Variational information maximization for feature selection. In Advances in Neural Information Processing Systems, pages 487–495.

[Garg et al., 2018] Garg, S., Galstyan, A., Ver Steeg, G., Rish, I., Cecchi, G., and Gao, S. (2018). Efficient representation for natural language processing via kernelized hashcodes. arXiv preprint arXiv:1711.04044.

[Grauman and Fergus, 2013] Grauman, K. and Fergus, R. (2013). Learning binary hash codes for large-scale image search. Machine learning for computer vision, 411(49-87):1.

[Hamamura et al., 2018] Hamamura, T., Suganuma, S., Ueda, M., Mearns, J., and Shimoyama, H. (2018). Standalone effects of a cognitive behavioral intervention using a mobile phone app on psychological distress and alcohol consumption among Japanese workers: Pilot non-randomized controlled trial. JMIR mental health, 5(1).

[Hausler, 1999] Hausler, D. (1999). Convolution kernels on discrete structures. Technical report.

[He et al., 2017] He, H., Balakrishnan, A., Eric, M., and Liang, P. (2017). Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. arXiv preprint arXiv:1704.07130.

[Joly and Buisson, 2011] Joly, A. and Buisson, O. (2011). Random maximum margin hashing. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pages 873–880. IEEE.

[Jozefowicz et al., 2016] Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., and Wu, Y. (2016). Exploring the limits of language modeling. arXiv preprint arXiv:1602.02410.

[Jurafsky and Martin, 2014] Jurafsky, D. and Martin, J. (2014). Dialog systems and chatbots. Speech and language processing, 3.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[Koeman and Heskes, 2014] Koeman, M. and Heskes, T. (2014). Mutual information estimation with random forests. In International Conference on Neural Information Processing, pages 524–531. Springer.

[Komorowski and Trzcinski, 2017] Komorowski, M. and Trzcinski, T. (2017). Random binary trees for approximate nearest neighbour search in binary space. arXiv preprint arXiv:1708.02976.
[Kosovan et al., 2017] Kosovan, S., Lehmann, J., and Fischer, A. (2017). Dialogue response generation using neural networks with attention and background knowledge.

[Kraskov et al., 2004] Kraskov, A., Stögbauer, H., and Grassberger, P. (2004). Estimating mutual information. Physical Review E, 69:066138.

[Kulis and Grauman, 2009] Kulis, B. and Grauman, K. (2009). Kernelized locality-sensitive hashing for scalable image search. In Computer Vision, 2009 IEEE 12th International Conference on, pages 2130–2137. IEEE.

[Lewinsohn et al., 1990] Lewinsohn, P. M., Clarke, G. N., Hops, H., and Andrews, J. (1990). Cognitive-behavioral treatment for depressed adolescents. Behavior Therapy, 21(4):385–401.

[Lewis et al., 2017] Lewis, M., Dauphin, Y. N., Parikh, D., and Batra, D. (2017). Deal or no deal? end-to-end learning for negotiation dialogues. arXiv preprint arXiv:1706.05125.

[Li et al., 2015] Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, B. (2015). A diversity-promoting objective function for neural conversation models. arXiv preprint arXiv:1510.03055.

[Li and Jurafsky, 2016] Li, J. and Jurafsky, D. (2016). Neural net models for open-domain discourse coherence. arXiv preprint arXiv:1606.01545.

[Li et al., 2017] Li, J., Monroe, W., Shi, T., Ritter, A., and Jurafsky, D. (2017). Adversarial learning for neural dialogue generation. arXiv preprint arXiv:1707.06547.

[Liu et al., 2016] Liu, C.-W., Lowe, R., Serban, I. V., Noseworthy, M., Charlin, L., and Pineau, J. (2016). How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. arXiv preprint arXiv:1603.08023.

[Lowe et al., 2017] Lowe, R., Noseworthy, M., Serban, I. V., Angelard-Gontier, N., Bengio, Y., and Pineau, J. (2017). Towards an automatic turing test: Learning to evaluate dialogue responses. arXiv preprint arXiv:1708.07149.

[Ly et al., 2017] Ly, K. H., Ly, A.-M., and Andersson, G. (2017). A fully automated conversational agent for promoting mental well-being: A pilot rct using mixed methods. Internet Interventions, 10:39–46.

[Moon et al., 2017] Moon, K. R., Sricharan, K., and Hero III, A. O. (2017). Ensemble estimation of mutual information. arXiv preprint arXiv:1701.08083.

[Mooney and Bunescu, 2005] Mooney, R. J. and Bunescu, R. C. (2005). Subsequence kernels for relation extraction. In Proc. of NIPS, pages 171–178.

[Morris et al., 2018] Morris, R. R., Kouddous, K., Kshirsagar, R., and Schueller, S. M. (2018). Towards an artificially empathic conversational agent for mental health applications: System design and user perceptions. Journal of medical Internet research, 20(6).

[Norouzi et al., 2014] Norouzi, M., Punjani, A., and Fleet, D. J. (2014). Fast exact search in hamming space with multi-index hashing. IEEE transactions on pattern analysis and machine intelligence, 36(6):1107–1119.

[Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting on association for computational linguistics, pages 311–318. Association for Computational Linguistics.

[Rasmussen, 2006] Rasmussen, C. E. (2006). Gaussian processes for machine learning.

[Ritter et al., 2010] Ritter, A., Cherry, C., and Dolan, B. (2010). Unsupervised modeling of twitter conversations. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 172–180. Association for Computational Linguistics.

[Schroeder et al., 2018] Schroeder, J., Wilkes, C., Rowan, K., Toledo, A., Paradiso, A., Czerwinski, M., Mark, G., and Linehan, M. M. (2018). Pocket skills: A conversational mobile web app to support dialectical behavioral therapy. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, page 398. ACM.

[Semeniuta et al., 2017] Semeniuta, S., Severyn, A., and Barth, E. (2017). A hybrid convolutional variational autoencoder for text generation. arXiv preprint arXiv:1702.02390.

[Serban et al., 2017a] Serban, I. V., Klinger, T., Tesauro, G., Talamadupula, K., Zhou, B., Bengio, Y., and Courville, A. C. (2017a). Multiresolution recurrent neural networks: An application to dialogue response generation. In AAAI, pages 3288–3294.

[Serban et al., 2016a] Serban, I. V., Lowe, R., Charlin, L., and Pineau, J. (2016a). Generative deep neural networks for dialogue: A short review. arXiv preprint arXiv:1611.06216.

[Serban et al., 2015] Serban, I. V., Lowe, R., Henderson, P., Charlin, L., and Pineau, J. (2015). A survey of available corpora for building data-driven dialogue systems. arXiv preprint arXiv:1512.05742.

[Serban et al., 2016b] Serban, I. V., Sordoni, A., Bengio, Y., Courville, A. C., and Pineau, J. (2016b). Building
end-to-end dialogue systems using generative hierarchical neural network models. In AAAI, volume 16, pages 3776–3784.

[Serban et al., 2017b] Serban, I. V., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A. C., and Bengio, Y. (2017b). A hierarchical latent variable encoder-decoder model for generating dialogues. In AAAI, pages 3295–3301.

[Shao et al., 2017] Shao, Y., Gouws, S., Britz, D., Goldie, A., Strope, B., and Kurzweil, R. (2017). Generating high-quality and informative conversation responses with sequence-to-sequence models. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2200–2209.

[Singh and Póczos, 2014] Singh, S. and Póczos, B. (2014). Generalized exponential concentration inequality for Rényi divergence estimation. In International Conference on Machine Learning, pages 333–341.

[Srivastava et al., 2013] Srivastava, S., Hovy, D., and Hovy, E. H. (2013). A walk-based semantically enriched tree kernel over distributed word representations. In Proc. of EMNLP, pages 1411–1416.

[Ver Steeg and Galstyan, 2014] Ver Steeg, G. and Galstyan, A. (2014). Discovering structure in high-dimensional data through correlation explanation. In Advances in Neural Information Processing Systems 27.

[Vinyals and Le, 2015] Vinyals, O. and Le, Q. (2015). A neural conversational model. ICML Deep Learning Workshop.

[Walters-Williams and Li, 2009] Walters-Williams, J. and Li, Y. (2009). Estimation of mutual information: A survey. In International Conference on Rough Sets and Knowledge Technology, pages 389–396. Springer.

[Wang et al., 2014] Wang, J., Shen, H. T., Song, J., and Ji, J. (2014). Hashing for similarity search: A survey. arXiv preprint arXiv:1408.2927.

[Wang et al., 2017] Wang, J., Zhang, T., Sebe, N., Shen, H. T., et al. (2017). A survey on learning to hash. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[Watanabe, 1960] Watanabe, S. (1960). Information theoretical analysis of multivariate correlation. IBM Journal of research and development, 4(1):66–82.

[Wen et al., 2016] Wen, T.-H., Vandyke, D., Mrksic, N., Gasic, M., Rojas-Barahona, L. M., Su, P.-H., Ultes, S., and Young, S. (2016). A network-based end-to-end trainable task-oriented dialogue system. arXiv preprint arXiv:1604.04562.

[Wu et al., 2017] Wu, Y., Wu, W., Xing, C., Zhou, M., and Li, Z. (2017). Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 496–505.

[Yu et al., 2017] Yu, L., Zhang, W., Wang, J., and Yu, Y. (2017). Seqgan: Sequence generative adversarial nets with policy gradient. In AAAI, pages 2852–2858.

[Zhai and Williams, 2014] Zhai, K. and Williams, J. D. (2014). Discovering latent structure in task-oriented dialogues. In ACL (1), pages 36–46.

[Zhao et al., 2014] Zhao, K., Lu, H., and Mei, J. (2014). Locality preserving hashing. In AAAI, pages 2874–2881.

A Derivations of the Information Theoretic Bounds

Before the discussion of our novel lower bound of mutual information, we introduce the information-theoretic quantity called Total Correlation (TC), which captures nonlinear correlation among the dimensions of a random variable C, i.e.,

$$TC(C) = \sum_j H(C_j) - H(C);$$

(4)

For a 2-D random variable, total correlation corresponds to mutual information quantity itself. And, $TC(C : Y)$ is defined as,

$$TC(C : Y) = TC(C) - TC(C|Y).$$

(5)

Intuitively, (5) describes the amount of information within C that can be explained by Y.

Along these lines, the mutual information quantity between the hashcodes can be decomposed as in Lemma below.

**Lemma 1** (Mutual Information Decomposition). *Mutual Information between $C^t$ and $C^p$ is decomposed as follows:*

$$I(C^t : C^p) = \sum_j I(C^t_j : C^p) - TC(C^t : C^p).$$

(6)
Proof.

\[ I(C^t : C^p) = \mathcal{H}(C^t) - \mathcal{H}(C^t | C^p) \]

\[ = \sum_j \mathcal{H}(C^t_j) - \sum_j \mathcal{H}(C^t_j | C^p) + \mathcal{H}(C^t) \]

\[ - \mathcal{H}(C^t | C^p) - \sum_j \mathcal{H}(C^t_j) + \sum_j \mathcal{H}(C^t_j | C^p) \]

\[ = \sum_j I(C^t_j : C^p) - (TC(C^t) - TC(C^t | C^p)) \]

\[ = \sum_j I(C^t_j : C^p) - TC(C^t : C^p). \]

Looking at the first term of RHS in (6), it is the mutual information between a one-dimensional and multi-dimensional random variable.

For these terms, since one of the variables is only 1-D, we can use the existing technique of variational bounds for an approximation, as in Lemma 2 below.

**Lemma 2.** Marginal mutual information for each bit in therapist hashcodes, \( I(C^t_j : C^p) \), is lower bounded as,

\[ I(C^t_j : C^p) \geq \mathcal{H}(C^t_j) + \langle \log q(C^t_j | C^p) \rangle_{p(C^t_j, C^p)}. \]  

(7)

Herein, \( \mathcal{H}(C^t_j) \) is easy to compute because \( C^t_j \) is a one-dimensional binary variable. For each of the proposal distributions \( q(C^t_j | C^p) \), we propose to use a Random Forest (RF) classifier [Gao et al., 2016].

In reference to the second term of RHS in (6), it is computationally intractable to compute the total correlation expression \( TC(C^t : C^p) \), which denotes the total correlations between bits of \( C^t \), explainable by \( C^p \). So, we would also like to obtain an upper bound of \( TC(C^t : C^p) \), which is cheap to compute, that would give us a lower bound for the second term in (6) because of the negative sign.

**Lemma 3.** \( TC(C^t : C^p) \) is upper bounded as:

\[ TC(C^t : C^p) \leq TC(C^t : Y^*) \]  

(8)

wherein \(|.|\) denotes the dimensionality of a random variable.

Although it is intractable to compute the original term \( TC(C^t : C^p) \), it is possible to compute \( TC(C^t : Y^*) \) for a latent variable representation \( Y^* \) of \( C^t \) that maximally explains the Total Correlations in \( C^t \).

We can think of the computation of the upper bound as an unsupervised learning problem. We propose to use an existing algorithm, CorEx, for the unsupervised learning of latent random variables representation \( Y^* \) [Ver Steeg and Galstyan, 2014].

It is important to note some practical considerations about the upper bound. In the case of a suboptimal solution to the maximization of \( TC(C^t : Y) \) above, the optimized quantity may not be an upper bound of \( TC(C^t : C^p) \), but rather an approximation. Also, the upper bound would not be tight if \( C^p \) doesn’t explain much of the total correlations in \( C^t \). Further, for even more computation cost reductions during the learning, the dimension of the latent representation \( Y \) can be kept much smaller than the dimension of hashcodes, i.e. \( |Y| \ll |C^p| \) for \( |C^p| \gg 1 \); this is because even a small number of latent variables should explain most of the total correlations for practical purposes as demonstrated by [Ver Steeg and Galstyan, 2014], and observed in our experiments on hashcodes as well.

Combining (7) and (8) into (6), we get the lower bound in Theorem 1.

Along same lines, we can derive the tight upper bound on joint entropy of hashcodes so as to obtain the normalization of the MI LB. From the definition of Total Correlation above (5), we have the following,

\[ \sum_j \mathcal{H}(C^t_j) - TC(C^t) = \mathcal{H}(C^t), \]

\[ TC(C^t) = TC(C^t : Y^*) + TC(C^t | Y^*), \]

and finally the expression below.

\[ \sum_j \mathcal{H}(C^t_j) - TC(C^t : Y^*) = \mathcal{H}(C^t) + TC(C^t | Y^*) \]

From this derived expression, we can simply obtain the upper bound and the corresponding gap.

**Previous Lower Bounds for Mutual Information:** Variational lower bounds on the mutual information criterion have been proposed in the past [Barber and Agakov, 2003, Chalk et al., 2016, Gao et al., 2016, Chen et al., 2016, Alemi et al., 2017, Garg et al., 2018]. Their lower bounds work only when one of the variables is fixed, say if \( C^t \) were fixed. In our objective, not only \( C^t \) is a functional of the hashing model that we are learning, it is high dimensional. Unless we have a lower bound for the entropy term \( \mathcal{H}(C^t) \) as well, which should be hard to obtain, we cannot use the above mentioned variational lower bounds for our problem as such. Besides, it is also non-trivial to find an appropriate proposal distribution \( q(C^t | C^p) \). Therefore, we adopt a different approach for obtaining a novel lower bound on the mutual information quantity, as described above.

**B Pseudo Code of Algorithms To Optimize LSH for Dialog Modeling**

In the following, we discuss the optimization of the reference set.
### Optimizing Reference Set.

For hashing models LSH-RKNN and LSH-RMM described in the main paper, we can optimize the reference set, \( S^R \), for dialog modeling as described in the following.

For the selection of elements in \( S^R \), we use a greedy algorithm maximizing the proposed mutual information lower bound (in Theorem [1]); see the pseudo code in Alg. [1]. We initialize a reference set of small size \( I \ll M \), by randomly selecting responses from the training set of patient/therapist responses, i.e. \( S = \{ S_1^p, \ldots, S_M^p, S_1^t, \ldots, S_N^t \} \); though, as noted before, the superset \( S \) for the random selection can be any set of sentences/paragraphs, not necessarily coming from a dataset of patient/therapist responses. First, each element in the initial reference set is optimized greedily, and then more elements are added one by one until the reference set size grows to \( M \). When optimizing each element in the set, for computing the MI lower bound, we sample \( \gamma \) number of response pairs from the training set of patient/therapist responses, \( \{(S_i^p, S_i^t)\}_{i=1}^N \). For computational efficiency, we adopt the idea of sampling for the candidate set as well, in each greedy optimization step, by sampling a subset of candidates of size \( \bar{\gamma} \) from the set \( S \).

The computation cost in the optimization is dominated by the number convolution kernel similarities, i.e. \( O(\gamma(M^2 + M \beta)) \). In practice, we can keep low values of \( \gamma \) as well as \( \beta \); in our experiments, we use \( \beta = 1000, \gamma = 100 \), and vary the value of \( M \) from 30 up to 300. A similar procedure can be used to optimize kernel parameters.

### Optimizing Neural Network Architecture.

For neural networks based LSH (LSH-RLSTM), we can optimize the number of layers and the units in each layer, by maximizing the proposed MI LB; see pseudo code in Alg. [2].

### C Experimental Details on LSH Models

LSTM models for each hash function in LSH-RLSTM are trained using Adam [Kingma and Ba, 2014], with the learning rate \( 1e-3 \), amsgrad=True, and \( l_1, l_2 \) regularization coefficients set to \( 1e-4 \). We initialize a word in a response and its POS tag with a random vector of size 30; for a single time step processing with in LSTM, word vectors of 10 adjacent words, along with their POS tags, are appended into a vector of size 600; this is required to avoid vanishing gradients since patient responses can be of length up to 8,000 words in the training dataset. For the \( H \) number of LSTM models as neural-hash functions, same neural architecture, i.e., same number of layers and units, are used in each model. When optimizing the architecture of the LSTM models with Alg. [2] by maximizing our proposed MI LB (MI LB Optimal), we add layers one by one greedily up to maximum possible 5 layers (\( L = 4, \gamma = 1000 \)), and try out different possible numbers of normal units in each layer, i.e., 4, 8, 16, 32, 64. (We keep the number of units small, since \( \alpha \) is small. Also, during the optimization, we use \( H = 30 \)). When optimizing the Reference set with Alg. [1] we keep \( \beta = 1000, \gamma = 100, I = 20 \).

### Similarity metric.

In case of kernel-hashing, we use subsequence kernels [Mooney and Bunescu, 2005] for computing similarity between two responses (subsequences of length up to 16 are matched), while similarity between a pair of words is computed as cosine between their word vector representations.

#### C.1 Subsequence Kernel

Let \( S_i \) and \( S_j \) be two sequences of tokens (words), the convolution kernel similarity between \( S_i \) and \( S_j \) is defined as below [Mooney and Bunescu, 2005].

\[
K(S_i, S_j) = \sum_{e=1}^{16} \sum_{|i| = |j| = e} \prod_{k=1}^{|i|} k(S_i(i_k), S_j(j_k)) \lambda^{l(i)+l(j)}.
\]

Here, \( k(S_i(i_k), S_j(j_k)) \) is the similarity between the \( k_{th} \) tokens in the subsequences \( i \) and \( j \), of equal length (up to value 16); \( l(.) \) is the actual length of a subsequence in the corresponding sequence, i.e., the difference between the end index and start index (subsequences do not have to be contiguous); \( \lambda \in (0, 1) \) is used to penalize the long subsequences. Dynamic programming is used for efficient computation of the subsequence kernel.

We use the following expression for computing the kernel similarity, \( k(a, b) \), between two tokens \( a \) and \( b \) using their respective word vector representations (real valued column vectors), \( w_a \) and \( w_b \).

\[
k(a, b) = cs \left( 1 - \frac{(1 - cs)}{(1 - \zeta)} \right); \quad cs = \frac{w_a^T w_b}{||w_a||_2 ||w_b||_2}, \quad (9)
\]

Herein, \( cs \) is cosine similarity between the two word vectors, \( w_a \) and \( w_b \); \( \lambda \) represents the positive part function; \( \zeta \in [-1, 1] \) is a parameter for compact support of the kernel function.

The parameters \( \lambda, \zeta \) are optimized by maximizing the proposed MI LB criterion in our dialog model.

### D Additional Experiment Details
Table 5: Textual responses generated by different systems for Depression Therapy dataset, along with ground truth patient and therapist responses.
Yeah, here’s hoping.
It’s going to keep on being a struggle.
There isn’t any, really anyone (ph) inside?
The Petition is the part that—
No, I don’t. I really don’t except that partly it’s a fear of being further condemned by her. I mean, she condemns me for everything I am and everything
I do.
So that there was again the concern with the - with [NAME].
Yeah. And so we didn’t talk all the way home at all. And as we didn’t talk, I think we were both growing angry. And so then we sat and we did talk
for a long time. And that was good. Well, it was good. It was more...we talked more deeply about ourselves and our feelings than we were in the past
without being angry at each other.
Yeah.
You’ve got your mother and sister.
Maybe if things are still horrible. I am hoping by next week they will be
When I told you that, did you remember what you felt?
And how to use that too.
You know people nearby pretty much though is you know? Also I think it’s also he cannot ... he’s been already like conditioned into, if he gets angry
or has any ill feelings towards certain members of the family that’s going to be it. Because he is no longer going be, you know ...
Well, that’s what we’re going to talk about.
From where?
I mean, you’ve been, I think it’s great, you going every week to see [NAME OTHER], just about?
What’s that, hon?
Yes.
Well yeah, like they didn’t want their kids to be the real dummies in the neighborhood (laughter). But it's not - my father, it’s not that he likes to brag
on his kids if their not there. I mean he’d never say anything nice about us in front of us. But I guess it’s the teachers that really bother me. I mean
why get a kid an answer book. I can’t - I mean, of course, I could have not cheated, but i And everything else I’ve done was very easy. My clarinet
lessons; [I had beef cattle] (ph); I was in girl scouts. And, I mean, all I had to do was pick them out. I never had do any work. The boys always did
the work, and then I went to the fair and got the purple ribbons.
Was it like defense-type stuff?
When you think of that moment in the car, how true do the words, "I can go on and thrive" feel on a 1-7 scale, with 1 being completely false and 7
being completely true?
I’m not seeing anybody right now. I’ve got to, I’ve got to worry about me first.
I don’t know. I thought of that all the time. Like how I’ve seen have big boxes of Kleenex usually expect the girls to cry.
It’s easier with the groups you so about.
And same at your church I think.
Yeah.
Okay.
And with that sort of a person. But because he is your biological father, there’s a certain obligation and you are kind enough to do what has to be
done. That for you to have feelings wouldn’t make much sense to me. Of course, where, where are the feelings to come from? The feelings would
come from the acts of kindness and caring, compassion. If they were not there, what should you feel? But you are, nevertheless, kind enough to do
the decent thing. So, so that’s one thing. But how do you feel about that? What I’ve just said?
I want a safe place out there.
No. I had it during finals my first year of grad school and during my second year probably for like four months towards the end.
Yeah, I get that.
Okay, that’s okay.
Once they send you the check -
It all seems so bad to you, it’s a disillusion.
Yeah.
I said you don’t want what I’m thinking on your tape.
So after I took one half at night, I was able to get up in the morning.
If I had asked you beforehand what they were going to do...?
You drink because you’re lonely.
That’s all right.
Yeah, in other words maybe I shouldn’t have gotten so upset. Whereas it wasn’t the most pleasant thing I should do. And like he was trying to say,
you know, can’t you just stay here tonight and leave tomorrow for college, see your girlfriend. And I’ve forgotten, it wasn’t until like I just got the
mail just before I came here and I was really, I didn’t want to open it. It was just two Valentine’s. It didn’t say anything, you know, except love, Daddy
and [NAME] and love, [NAME], but...
You went out
Um-hum.
Let’s just pretend for a moment. Let’s pretend that I’m your boss.
Okay. How old are you now?
Right. Well, a lot of it was like, before Christmas I was really, really snowed under with stuff and I had to work every day and I had to work late
almost every day, and somehow even just the hour coming here was really a lot. Plus, I was really functioning on, ‘let me just get through with work
because I had so many hours that I had to get through by Christmas or I couldn’t go home for Christmas, and it was sort of, ‘let me just get through
that and think about things later’.
I was starting, you know, itching, it’s almost for a month, a month and a half, I keep itching a lot over here in my face.
I feel like I’ve lost credentials and hope over...
Will you try out something again right now what you would find to be a response to Shelley given the fact of last week for example?
The rapport was good enough for that.

Table 6: We show first 50 responses (in raw text format) that are selected greedily for the reference set (S^R) in LSH-RkNN model, out of 76,000 responses from the train set, with Alg. [I]
Algorithm 1 Optimizing Reference Set in LSH-RkNN or LSH-RMM for Dialogue Modeling

Require: Training set $S = \{S^p, S^t\}$; initial and final size of reference set, $I$ and $M$ respectively; $\beta$ and $\gamma$ are the number of samples, as candidates for the reference set, and for computing the lower bound, respectively.

1: $r^d \leftarrow \text{randomSubset}(2N, I)$  % random subset of indices, of size $I$, from $\{1, \cdots, 2N\}$ for initialization of reference set
   % optimizing the reference set up to size $M$ greedily
2: for $j = 1 \rightarrow M$ do
3:   if $j > I$ then
4:     $r^d \leftarrow \{r^d, \text{randomSubset}(2N, 1)\}$  % adding one more element in the reference set, that is to be optimized
5:   end if
6:   $r^{ref} \leftarrow \text{randomSubset}(2N, \beta)$  % subset of the structures as candidates for the reference set
7:   $r^{lb} \leftarrow \text{randomSubset}(N, \gamma)$  % subset of patient/therapist responses pairs for computing the MI lower bound
8:   $K^p \leftarrow \text{computeKernel}(S^p(r^{lb}), S^{ref}(r^d))$  % $\gamma \times j$ size
9:   $K^t \leftarrow \text{computeKernel}(S^t(r^{lb}), S^{ref}(r^d))$  % $\gamma \times \beta$ size
10:  $\bar{K}^p \leftarrow \text{computeKernel}(S^p(r^{lb}), \bar{S}(r^{ref}))$  % $\gamma \times j$ size
11:  $\bar{K}^t \leftarrow \text{computeKernel}(S^t(r^{lb}), \bar{S}(r^{ref}))$  % $\gamma \times \beta$ size
12:  $\text{mi}^{lb} \leftarrow \text{computeMILowerBound}(K^p, K^t, \bar{K}^p, \bar{K}^t)$  % compute the MI lower bound for all the candidates $\bar{S}(r^{ref})$, using the kernel matrices, via computation of hashcodes
13:  $r^{d}(j) \leftarrow \text{maxMILBDataIndex}(\text{mi}^{lb}, r^{ref})$  % choose the index with maximum value of MI lower bound, from the set of candidate indices $r^{ref}$
14: end for
15: return $\bar{S}(r^d)$

Algorithm 2 Optimizing Neural Architecture in LSH-RLSTM for Dialog Modeling

Require: Training dataset $\{S^p, S^t\}$, with $N$ pairs; maximum number of layers in neural language models, $L$; the number of samples for computing the MI lower bound, $\gamma$; values for units in a layer, $u = \{4, 8, 16, 32, 64, \text{None}\}$.
   % optimizing up to $L$ layers greedily
1: for $j = 1 \rightarrow L$ do
2:   $r^{lb} \leftarrow \text{randomSubset}(N, \gamma)$  % subset of patient/therapist responses pairs for computing the MI LB to optimize $j_{th}$ layer
3:   for $l \in u$ do
4:     $n(j) \leftarrow l$  % $l$ units in $j_{th}$, layer of neural network
5:     $C^p \leftarrow \text{computeHashcodes}(S^p(r^{lb}), n)$
6:     $C^t \leftarrow \text{computeHashcodes}(S^t(r^{lb}), n)$
7:     $\text{mi}_{lb}(l) \leftarrow \text{computeMILowerBound}(C^p, C^t)$
8: end for
9: $n(j) \leftarrow \text{maxMILBIndex}(\text{mi}_{lb}, u)$  % choose the units with maximum value of MI LB
10: if $n(j)$ is None then break out of loop end if
11: end for
12: return $n$