

# Dialogue Modeling Via Hash Functions

Sahil Garg<sup>1</sup>, Guillermo Cecchi<sup>2</sup>, Irina Rish<sup>2</sup>, Shuyang Gao<sup>1</sup>,  
Greg Ver Steeg<sup>1</sup>, Sarik Ghazarian<sup>1</sup>, Palash Goyal<sup>1</sup>, Aram Galstyan<sup>1</sup>

<sup>1</sup> USC Information Sciences Institute, Marina del Rey, CA USA

<sup>2</sup> IBM Thomas J. Watson Research Center, Yorktown Heights, NY USA

sahil.garg.cs@gmail.com, {gcecchi,rish}@us.ibm.com, sgao@isi.edu,  
{gregv,ghazarian,goyal,galstyan}@isi.edu

## Abstract

We propose a novel machine-learning framework for dialogue modeling which uses representations based on hash functions. More specifically, each person’s response is represented by a binary hashcode where each bit reflects presence or absence of a certain text pattern in the response. Hashcodes serve as compressed text representations, allowing for efficient similarity search. Moreover, hashcode of one person’s response can be used as a feature vector for predicting the hashcode representing another person’s response. The proposed hashing model of dialogue is obtained by maximizing a *novel lower bound* on the *mutual information* between the hashcodes of consecutive responses. We apply our approach in psychotherapy domain evaluating its effectiveness on a real-life dataset consisting of therapy sessions with patients suffering from depression; in addition, we also model transcripts of interview sessions between Larry King (television host) and his guests.

## 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 challenging open problem, developing dialogue systems for particular applications can be more tractable, due to specific properties of the application domain.

A motivating application for our work is (semi-)automated psychotherapy: easily accessible, round-the-clock psychotherapeutic services provided by a conversational agent. According to recent estimates, mental health disorders affect one in four adult Americans, one in six adolescents, and one in eight children. Furthermore, as predicted by the World Health Organization, by 2030 the amount of worldwide disability and life loss attributable to depression may become greater than for any other condition, including cancer, stroke, heart disease, accidents, and war. However, many people do not receive an adequate treatment; one of the major factors here is limited availability of mental health professionals, as

---

*Its all my fault. It’s like getting hit in the stomach And trying to catch your breath. You can’t catch it. You struggle, move around, You just can’t get your breath. That just really knocks me off balance. You begin to generate this feeling of a kind of negativism. And that’s why it really being hurt. I never allowed myself to think negative, accept everybody else. I am the only one giving back to the “be quiet time.” I had to have something to make do with that.*

0101001111001011111110101101011110011011100011101

*OK. Maybe that fits into the whole pattern then.*

011100110100101111110001111111110001010110011001

---

Table 1: In this table, we show pairs of patient and therapist (Blue) responses. For each response, we obtain a locality sensitive hashcode as its representation. The hashcode of a patient response is used as feature representation to infer the hashcode of the corresponding therapist response.

compared to the number of potential patients. Thus, creating automated systems providing mental health services is becoming an increasingly important area of research and development; however, existing conversational systems in this domain are still primarily rule-based, and lack sophisticated dialogue modeling and generation capabilities based on actual text understanding.

An interesting feature of therapeutic dialogues appears to be a classical pattern of a relatively long monologue of a patient followed by a considerably shorter response of a therapist, e.g., a typical example is shown in Tab. 1 (taken from a dataset we experimented with, containing multiple transcribed depression therapy sessions.) 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, more constrained response can be a somewhat easier task than solving the open-domain dialogue challenge. Note that, 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.

Motivated by above considerations, we introduce here a novel dialogue modeling framework where responses are represented as *locality-sensitive binary hashcodes* [Kulis and Grauman, 2009; Joly and Buisson, 2011; Garg *et al.*, 2018b]. The motivation behind such approach includes several con-

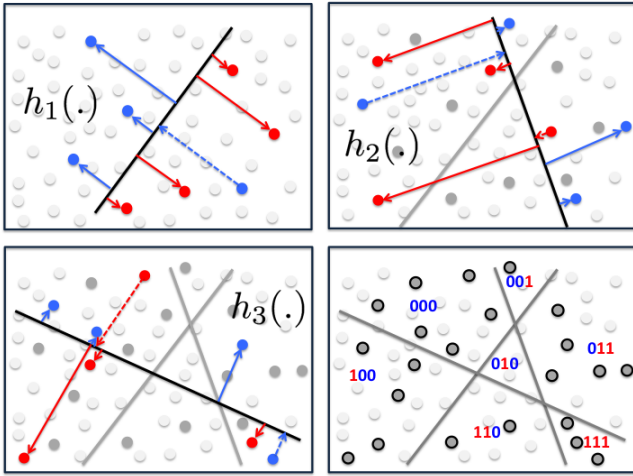


Figure 1: An illustration of the locality sensitive hashing approach. In the figure, each dialogue response is represented as a dot. In the top left of the figure, for building a hash function, 4 responses are selected randomly for assignment of artificial red color labels, representing hash bit value 1, and another 4 responses selected randomly for assignment of blue labels. Having these 8 responses with artificially assigned labels, we can learn a maximum margin boundary as hash function  $h_1(\cdot)$ , i.e. training a kernel-SVM model discriminating between the red and blue labeled dialogue responses; instead of a maximum margin boundary, one can learn a kernel or neural language model based (regularized) boundary as well. Following same procedure, we assign artificial labels to other small subsets of responses, so as to learn hash functions  $h_2(\cdot)$ ,  $h_3(\cdot)$ , and so on. In the bottom right of the figure, we see hashcodes assigned to responses as per the three hash functions; the dark gray dots represent those responses which we used for building the hash functions, referred as a reference set; for computing hashcode of a given test response (light gray dot), we need to compute its kernel similarities w.r.t. the dark gray dots only. In practice, we first select the reference set (dark gray dots), as a random or optimized subset of all the responses (all the dots), and then take smaller random subsets (red or blue labels) out of the reference set for building hash functions.

siderations: (a) it can be easier to infer a generalized, interpretable representation (e.g., binary hascode) of a response (along with the inverse mapping) instead of a specific textual response (due to potentially smaller search space); (b) hashcodes of patient responses can be used as feature vectors to infer the hashcodes of therapist responses (multiple binary class labels), as previously demonstrated for an information-extraction task [Garg *et al.*, 2018a]; (c) an inferred therapist’s hashcode can be efficiently mapped to multiple text sentences (possible choices for the therapist response). See Tab. 1.

While the above approach was motivated by psychotherapy domain, it is generally applicable to a wide range of other domains involving dialogue modeling and generation. The *key contributions* of this work include:

- a novel, generic framework for modeling a dialogue using *locality sensitive hash functions* (based on kernel similarities or neural networks, as discussed later) to represent textual responses;
- a *novel lower bound on the Mutual Information (MI)* between the hashcodes of the responses from the two

agents (capturing relevance between the two), used as an optimization criterion to optimize a hashing model;

- an approach using the MI lower bound as a metric for joint evaluation of the representation quality of hashcodes, and the quality of inference. Also, a tight *upper bound on joint entropy* is derived, to separately characterize the quality of hashcodes as representations.

Empirical evaluation of the proposed framework and the optimization approach, which uses both kernels and neural networks for locality sensitive hash functions, is performed on a dataset containing textual transcriptions of 400 depression-therapy sessions conducted by human therapists with real patients. We find that our approach of optimizing hashing by maximizing our novel MI lower bound provides a consistent improvement over alternative types of hashing approaches (using both kernels and neural nets), in terms of the proposed evaluation metrics. Further, general applicability of our work is demonstrated by experiments on another dataset of transcriptions of 75 interview sessions between television host Larry King and his guests.

## 2 Related Work

Dialogue agents, e.g. chat bots, are becoming increasingly popular in various applications, including (semi)-automated psychotherapy, for example, based on popular techniques such as CBT (Cognitive Behavioral Therapy [Lewinsohn *et al.*, 1990]); however, these agents have very limited abilities to actually understand free text responses from their users; instead, they are typically offering a fixed set of pre-programmed responses to choose from [Fitzpatrick *et al.*, 2017]. See [Jurafsky and Martin, 2014] for an overview.

There are several neural networks based approaches proposed in the recent past for dialogue modeling in general domain [Serban *et al.*, 2015; 2016; 2017a; 2017b; Shao *et al.*, 2017; Asghar *et al.*, 2017; Wu *et al.*, 2017]. However, the setting is somewhat different from the therapy dialogues where the patient responses can be extremely long (up to tens of thousands of words). 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. Also, therapy dialogue modeling and generation has some similarities with the task-driven dialogue [Zhai and Williams, 2014; Wen *et al.*, 2016; Althoff *et al.*, 2016; Lewis *et al.*, 2017; He *et al.*, 2017], although evaluating effectiveness of a therapeutic dialogue may be more challenging as the effect is not always immediate. Attention to specific parts of the response, as well as background knowledge, explored in neural network based dialogue modeling [Kosovan *et al.*, ] can be quite helpful in therapeutic dialogues; those aspects are, to some extent, implicitly captured by learning the hash models. Note that in related work by [Bartl and Spanakis, 2017], hashcodes are not treated as representations of the responses, and used only for the nearest neighbor search, unlike the approach proposed here.

---

**Algorithm 1** Generalized Algorithm for Constructing Locality Sensitive Hash Functions, with Kernel Functions or Neural Networks

---

**Require:** Reference set of structures  $\mathcal{S}^R$ , of size  $M$ ; sub-sampling parameter,  $\alpha$ , for randomizing hash function; the number of hash functions,  $H$ . % Random subsets, fixed across structure inputs  
1:  $\mathbf{z} \leftarrow \{\mathbf{0}_\alpha, \mathbf{1}_\alpha\}$  % labels for hash code bit  
2: **for**  $j = 1 \rightarrow H$  **do**  
3:  $\mathbf{r}_j^h \leftarrow \text{randomSubset}(M, 2\alpha)$  % Random subsampling of structures for building a randomized locality sensitive hash function  
4:  $\mathcal{H}_j \leftarrow \text{TrainBinaryClassifierAsHashFunction}(\mathcal{S}_R(\mathbf{r}_j^h), \mathbf{z})$  % Train a binary classifier, on the random subset of structures  $\mathcal{S}_R(\mathbf{r}_j^h)$  with their randomly assigned hash labels  $\mathbf{z}$ , either using a convolution kernel similarity (subsequence/path/graph kernels) based classification model such as, SVM, kNN, or a neural language model, such as LSTM, GRU, serving as a binary hash function  
5: **end for**

---

The idea of maximizing mutual information objective for dialogue modeling has been considered previously by [Li *et al.*, 2015], though it was only used in test setting, rather than as a guide for training.

An evaluation metric such as BLEU score [Papineni *et al.*, 2002] may not be the best for our application, it tries to capture *all information* in text when comparing an inference w.r.t. the ground truth, rather than evaluate the relevance of one response to the other. We propose several evaluation metrics based on binary hashcodes themselves, since the latter are assumed to serve as efficient representations of text in our dialogue context.

### 3 Dialogue Modeling via Binary Hashing

Here we discuss our novel framework for modeling a dialogue between two agents using binary hash functions. In our formulation below, we refer to the two agents as a patient and a therapist, respectively, although the approach can be clearly applied to a variety of other dialogue settings.

#### 3.1 Problem Formulation

In a dialogue session between a patient and a therapist, we denote  $i_{th}$  response from the patient as  $S_i^p$ , and the corresponding response from the therapist as  $S_i^t$ ; in a session, we have  $\mathcal{S}^{pt} = \{S_i^p, S_i^t\}_{i=1}^N$ , with additional notations for sets,  $\bar{\mathcal{S}} = \{S_1^p, \dots, S_N^p, S_1^t, \dots, S_N^t\}$ ,  $\mathcal{S}^p = \{S_1^p, \dots, S_N^p\}$ ,  $\mathcal{S}^t = \{S_1^t, \dots, S_N^t\}$ . In this work, we consider a response  $S_i$ , be it from the patient or the therapist in a session, as a natural language structure which can be simply plain text, or text along with part of speech tags (PoS), or syntactic/semantic parsing of the text. As per a typical dialogue modeling setup, the task would be to predict  $S_i^t$  given  $S_i^p$ , i.e. generating therapist responses. However, we propose a different setup. We propose to encode each response of a patient/therapist ( $S_i^p/S_i^t$ ) in a therapy session as a binary hashcode ( $\mathbf{c}_i^p/\mathbf{c}_i^t \in \{0, 1\}^H$ ), and focus upon the problem of inferring the therapist hashcode ( $\mathbf{c}_i^t$ ) given the hashcode ( $\mathbf{c}_i^p$ ) of the patient response ( $S_i^p$ ), *before finally mapping the inferred therapist hashcode ( $\mathbf{c}_i^{t*}$ ) to multiple text response choices*. Note that  $\mathbf{c}_i^t$  is the hashcode representation of the ground truth therapist response  $S_i^t$  (so it is the groundtruth itself), whereas  $\mathbf{c}_i^{t*}$  is the inference given only the knowledge of  $S_i^p$ , its hashcode  $\mathbf{c}_i^p$ , and no knowledge of  $S_i^t$ ; all the hashcodes are generated using the same hashing model  $\mathcal{M}_h$ . See Tab. 1 for examples of pairs of patient/therapist responses and their corresponding hashcode representations.

Note that, herein, we keep problem setting simple, considering only the previous response of the patient to make an inference of the therapist response, although the proposed approach of modeling responses via hash functions is generic enough for more sophisticated dialogue modeling setups such as the ones using reinforcement learning.

A hashcode for a dialogue response can be thought of as a generalized representation of the response such that the binary bit values in the hashcode denote existence of important patterns in a response which are relevant for the therapeutic dialogue between the patient and the therapist whereas the rest in the response is effectively noise that should be irrelevant from psychotherapy perspective. The additional advantage of modeling responses as hashcodes is that we can map a binary hashcode to multiple text responses efficiently, i.e. generating therapist responses as per the inferred therapist hashcode.

Next, we describe our approach for generating locality sensitive hashcodes of dialogue responses, using either kernel similarity functions, or neural models such as LSTM-s.

#### 3.2 Locality Sensitive Hashcodes of Responses using Kernel Functions or Neural Language Models

The main idea behind locality sensitive hashing is that data points that are *similar* to each other should be assigned hashcodes with minimal Hamming distance to each other, and vice versa. A similarity/distance function need not to be defined explicitly; see [Wang *et al.*, 2014] for details.

Since locality sensitive hashing ensures that natural language structures, that are assigned hashcodes 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)<sup>1</sup>, and so for the responses in a dialogue.

Many hash functions have been proven to be locality sensitive, with rigorous mathematical proofs [Wang *et al.*, 2017]. However, the literature on data driven locality sensitive hash functions is recent, such as based on kernel similarity functions [Kulis and Grauman, 2009; Joly and Buisson, 2011; Garg *et al.*, 2018b]; see Fig. 1 for a demonstration of locality sensitive hashing of dialogue responses. Although these

---

<sup>1</sup>A similarity function doesn't imply a semantic similarity function here. For instance, as per a learned locality sensitive hashing model, the implied (valid) similarity function may account for matching of only certain patterns in textual responses.

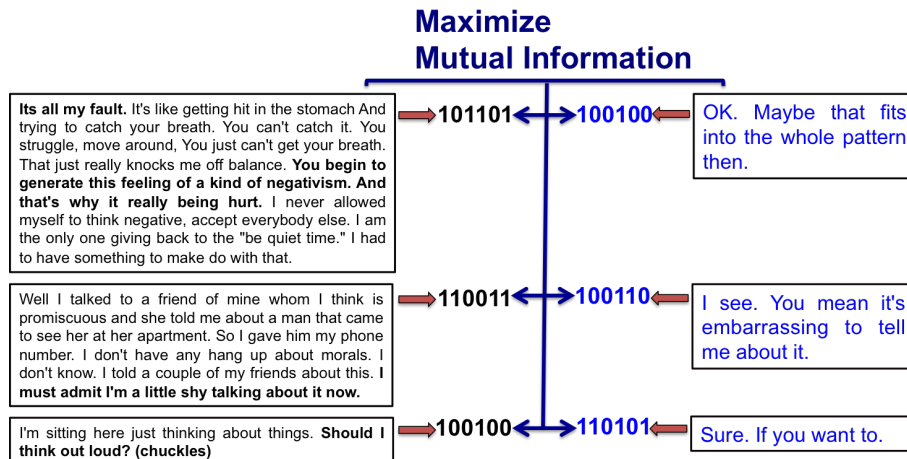


Figure 2: In left side of the figure, we have patient textual responses from depression dataset, and the corresponding therapist responses (in blue colored text) are shown on the right. We optimize a hashing model for dialogue modeling so that Mutual Information (MI) between hashcodes of consecutive (textual) responses is maximized in a training dataset; our intuition is that, with maximization of MI, hashcodes should encode textual information in responses that is mutually relevant for the conversation, while ignoring the rest as noise. For instance, we would like a good hashing model to encode the bold text patterns in the patient responses, since those are highly relevant for the therapist responses. Note that we maximize MI only for training of hashing model, not for the inference of therapist hashcodes.

works lack theoretical guaranties for locality sensitivity of hash functions, the common intuitive principle in these approaches is to learn a randomized binary hash function, constructed using a very small subset of available data points. For instance, in [Joly and Buisson, 2011], a kernelized hash function is built by learning a (random) maximum margin boundary (SVM) that discriminates between the samples of two small random subsets of a super set; same principle is followed in [Garg *et al.*, 2018b] for constructing a random hash function, i.e. obtaining a k-Nearest Neighbors based discrimination boundary between the two random subsets using kernel functions, instead of a maximum margin boundary. In [Kulis and Grauman, 2009; 2012], a hash function is built using the union of two random subsets, that is an (approximately) random linear hyperplane in the kernel implied feature space. Despite the lack of theoretical guaranties for locality sensitivity of kernel based hash functions, the principle of constructing each random hash function by learning it on a very small random subset of samples works well in practice.

Considering the commonality in these hashing methods due to the principle of random sub-sampling, we present a generalization in Alg. 1 for generating locality sensitive hashcodes, that is applicable for building a random hash function, operating on language structures, either (i) using a convolution kernel,  $K(S_i, S_j)$ , defining similarity between two structures  $S_i, S_j$  [Mooney and Bunescu, 2005; Collins and Duffy, 2002; Haussler, 1999], as proposed in the previous works [Kulis and Grauman, 2009; Joly and Buisson, 2011; Garg *et al.*, 2018b], or *as we additionally propose* in this paper, (ii) using a neural language model such as LSTM, GRU, etc.

In Alg. 1, the pseudo code is generically applicable for obtaining hashcode of any language structure, including responses from a dialogue. We have a set of language structures,  $S^R$  of size  $M$ , with no class labels. For building  $j$ th hash function, we take a random subset  $S^R(r_j^h)$  of size  $2\alpha$ ,

s.t.  $1 < \alpha \ll M$ , from the set  $S^R$ ; this subset is partitioned into two subsets, each of size  $\alpha$ , through the assignment of artificially generated hash bit labels  $z$ . Having the two small subsets, we can learn a binary classifier (preferably regularized), based on kernel similarities or a neural language model, that discriminates between the two subsets, acting as a hash function. Following this principle, we can learn  $H$  number of hash functions, with each one taking constant number of computations for learning since the size of the subset  $S^R(r_j^h)$  is a small constant,  $2\alpha \ll M$ .

When building the hash functions using kernel similarity functions, the size  $M$  of the reference set  $S^R$  should be small. This is because generation of a kernelized hashcode for an input structure  $S_i$  using the locality sensitive hashing algorithm, requires computing convolution kernel similarities of  $S_i$  with all the elements of the reference set  $S^R$ . A hashcode  $c_i$  for a structure  $S_i$  represents finding important substructures in it related to the set  $S^R$ . This means, when generating hashcodes of dialogue responses as representations in our framework, we can control what patterns should be attended by the therapist in the patient responses through the optimization of the reference set itself. This idea can be very powerful when modeling dialogues involving long responses from patients. In Sec. 4, we propose to optimize a reference set using a novel algorithm.<sup>2</sup>

On the other hand, when constructing a parametric hash function by learning a binary classifier based on a neural language model as proposed above, *we don't need to restrict the size of reference set,  $S^R$* . Instead, we propose to optimize

<sup>2</sup>Note that the similarity function, implied from a kernel locality sensitive hashing model, may not be same as the kernel function. While the kernel function may be accounting for similarity of all possible patterns in textual responses, the similarity function would be effectively matching only certain patterns, defined as per the reference set in the hashing model.

the architecture of all the  $H$  number of neural models, corresponding to  $H$  hash functions, jointly with our novel algorithm in Sec. 4 while learning the weight parameters of the neural models independently of each other.

### 3.3 Mapping Inferred Hashcodes to Textual Responses

While we propose to represent patient/therapist responses as hashcodes, and infer the hashcode for a therapist response instead of directly inferring the textual response, we can map an inferred therapist hashcode to multiple choices for the final textual response. As mentioned previously, one of the main advantages of using binary hash functions to build representations of responses is that one can use the same binary hashing model for indexing a repository of sentences. For the task of building an automated therapist (assisting a human therapist), from a training set of therapy sessions, we can put all the responses from therapists into a single repository. Then, having a hashing model as per Alg. 1, all the responses can be indexed, with binary hashcodes. Now, after inferring a therapist hashcode, we can search in the repository of hashcode-indexed responses, to find multiple responses having minimal hamming distance w.r.t. the inferred hashcode. Even if the number of hashcode bits were large, sub-linear time algorithms can be used for efficient similarity search (nearest neighbors) in hamming spaces [Norouzi *et al.*, 2014; Komorowski and Trzcinski, 2017]. For e.g., see Tab. 4.

## 4 Learning Hashing for Dialogue Modeling with Mutual Information Maximization

In the previous section, we introduced a framework for modeling dialogues via hash functions, especially in the context of a therapeutic dialogue between a therapist and a patient. The question is what objective function to use for the optimization of the hashing model? In the following, we introduce an objective function that is suitable for optimizing hash functions for dialogue modeling. The two main criteria for selecting this objective function are as follows: (1) it should characterize the quality of hashcodes as generalized representations of dialogue responses; (2) it should also account for the inference of therapist hashcodes, i.e. therapist dialogue generation. Although the primary motivation of the present work is therapeutic dialogue modeling, we note that the above criteria also apply to more general dialogue modeling problems as well.

### 4.1 Objective Function Formulation to Optimize Hashing for Dialogue Modeling

We propose that maximizing *Mutual Information* between the hashcodes of patient and the corresponding therapist dialogue responses is suitable to optimize on both the inference accuracy as well as the representation quality.

$$\arg \max_{M_h} \mathcal{I}(\mathcal{C}_p : \mathcal{C}_t; M_h) \quad (1)$$

$$\mathcal{I}(\mathcal{C}_p : \mathcal{C}_t; M_h) = \underbrace{\mathcal{H}(\mathcal{C}_t; M_h)}_{\text{representation}} - \underbrace{\mathcal{H}(\mathcal{C}_t | \mathcal{C}_p; M_h)}_{\text{inference}} \quad (2)$$

---

**Algorithm 2** Algorithm for Optimizing Neural Architecture in a Neural-Hashing Model for Dialogue Modeling by Maximizing Our MI LB

---

**Require:** Train sets,  $\mathcal{S}^p, \mathcal{S}^t$ ; 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,  $\mathbf{u} = \{5, 10, 20, 40, \text{None}\}$ .  
*% optimizing up to L layers greedily*  
1: **for**  $j = 1 \rightarrow L$  **do**  
2:  $\mathbf{r}^{lb} \leftarrow \text{randomSubset}(N, \gamma)$  *% subset of patient/therapist responses pairs for computing the MI LB to optimize  $j_{th}$  layer*  
*% Number of units in  $j_{th}$  layer, None not applicable for 1<sup>st</sup> layer*  
3: **for**  $l \in \mathbf{u}$  **do**  
4:  $\mathbf{n}^{(j)} \leftarrow l$  *% l units in  $j_{th}$  layer of neural network*  
5:  $\mathcal{C}^p \leftarrow \text{computeHashcodes}(\mathcal{S}^p(\mathbf{r}^{lb}), \mathbf{n})$   
6:  $\mathcal{C}^t \leftarrow \text{computeHashcodes}(\mathcal{S}^t(\mathbf{r}^{lb}), \mathbf{n})$   
7:  $\mathbf{mi}_{lb}(l) \leftarrow \text{computeMILowerBound}(\mathcal{C}^p, \mathcal{C}^t)$   
8: **end for**  
9:  $\mathbf{n}^{(j)} \leftarrow \text{maxMILBIndex}(\mathbf{mi}_{lb}, \mathbf{u})$  *% choose the units with maximum value of MI LB*  
10: **if**  $\mathbf{n}^{(j)}$  is None **then** break out of loop **end if**  
11: **end for**  
12: **return**  $\mathbf{n}$

---

Herein,  $\mathcal{M}_h$  denotes a hashing model, such as the one described above;  $\mathcal{C}_p$  is the distribution on the hashcodes of patient dialogue responses, and  $\mathcal{C}_t$  characterizes the distribution on the hashcodes of the corresponding therapist responses. While minimizing the conditional entropy,  $\mathcal{H}(\mathcal{C}_t | \mathcal{C}_p; M_h)$ , is to improve the accuracy for the inference of the therapist response hashcodes, maximizing the entropy term,  $\mathcal{H}(\mathcal{C}_t; M_h)$ , should ensure good quality of the hashcodes as generalized representations. See Fig. 2 for an illustration.

Computing mutual information between two high-dimensional variables can be expensive, potentially inaccurate if the number of samples is small [Kraskov *et al.*, 2004]. So, we propose to optimize the hashing model, by maximizing a lower bound on the mutual information criterion.

### 4.2 Information Theoretic Bounds for Optimizing and Evaluating Hashing for Dialogue Modeling

We develop a novel lower bound on the mutual information criterion, that is cheaper and more accurately computable than the criterion itself.

**Theorem 1** (Lower Bound for Mutual Information). *Mutual information between two hashcode distributions,  $\mathcal{I}(\mathcal{C}_t : \mathcal{C}_p; M_h)$ , is lower bounded as,*

$$\mathcal{I}(\mathcal{C}_t : \mathcal{C}_p; M_h) \geq \sum_j \mathcal{H}(\mathcal{C}_t(j); M_h) - \mathcal{TC}(\mathcal{C}_t : \mathcal{Y}^*; M_h) + \sum_j \langle \log q(\mathcal{C}_t(j) | \mathcal{C}_p; M_h) \rangle_{p(\mathcal{C}_t(j), \mathcal{C}_p; M_h)}. \quad (3)$$

$$\mathcal{Y}^* \leftarrow \arg \max_{\mathcal{Y}: |\mathcal{Y}|=|\mathcal{C}_p|} \mathcal{TC}(\mathcal{C}_t : \mathcal{Y}; M_h) \quad (4)$$

Herein,  $\mathcal{TC}(\mathcal{C}_t : \mathcal{Y}; M_h)$  describes the amount of Total Correlations (Multi-variate Mutual Information)<sup>3</sup> within  $\mathcal{C}_t$  that

<sup>3</sup>The ‘‘total correlation’’ quantity, also called as multi-variate Mutual Information, or multi-information, was defined in [Watanabe, 1960; Studený and Vejnarová, 1998; Kraskov *et al.*, 2005].



---

**Algorithm 3** Algorithm for Optimizing Reference Set in Kernelized Hashing for Dialogue Modeling by Maximizing Our MI LB

---

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

---

can be explained by a latent variables representation  $\mathcal{Y}$ ;  $q(\mathcal{C}_t(j)|\mathcal{C}_p; M_h)$  is a proposal conditional distribution for the  $j_{th}$  bit of therapist hashcodes built using a classifier, like a random forest, neural network, etc.

An interesting aspect of the quantity  $\mathcal{TC}(\mathcal{C}_t : \mathcal{Y}; M_h)$  is that one can compute it efficiently for optimized  $\mathcal{Y}^*$  that explains maximum possible Total Correlations present in  $\mathcal{C}_t$ , see [Ver Steeg and Galstyan, 2014] for more details on the optimization; for practical purposes, the dimension of latent representation  $\mathcal{Y}$  can be kept much smaller than the dimension of hashcodes, i.e.  $|\mathcal{Y}| \ll |\mathcal{C}_p|$  for  $|\mathcal{C}_p| \gg 1$ .

As we observed for the mutual information criterion above, we can also see different terms in the mutual information lower bound having varying roles for the optimization; for instance, the first two terms in (3) contribute to improve the quality of hashcodes as representations, i.e. maximizing entropy of each hashcode bit while discouraging redundancies between the bits, and the last term of conditional entropies is for improving inference of hashcode bits individually.

We argue to use the proposed MI lower bound, for not only optimizing hashing for dialogue modeling, but also as an evaluation metric quantifying the representation quality as well as the inference of hashcodes; besides, it serves to quantify what is mutually relevant between the responses of two dialog agents.

Further, for evaluating representation quality of hashcodes separately, in Theorem 2 below, we present a novel tight upper bound for joint entropy of hashcodes.

**Theorem 2** (Upper Bound for Entropy). *Joint entropy of hashcodes,  $\mathcal{H}(\mathcal{C}_t; M_h)$ , is upper bounded as,*

$$\mathcal{H}(\mathcal{C}_t; M_h) \leq \sum_j \mathcal{H}(\mathcal{C}_t(j); M_h) - \mathcal{TC}(\mathcal{C}_t : \mathcal{Y}^*; M_h);$$
$$\mathcal{Y}^* \leftarrow \arg \max_{\mathcal{Y}: |\mathcal{Y}|=|\mathcal{C}_p|} \mathcal{TC}(\mathcal{C}_t : \mathcal{Y}; M_h)$$

Note the similarities between the expressions for proposed the mutual information lower bound, and the joint entropy upper bound; the first two terms match between the two expressions.

For  $\mathcal{TC}(\mathcal{C}_t|\mathcal{Y}^*; M_h) = 0$ , i.e. when a latent representation  $\mathcal{Y}^*$  is learned which explains all the total correlations in  $\mathcal{C}_t$ , the upper bound becomes equal to the entropy term; practically, for the case of hashcodes, learning such a representation should not be too difficult, so the bound should be very tight. This is quite interesting as we are able to judge the representation quality of hashcodes, that we proposed to quantify via the entropy of hashcodes in the above, through the tight upper bound in Theorem 2 which is much easier and cheap/accurate to compute than the entropy term itself. Besides the mutual information lower bound, for more detailed analysis, we propose to use the upper bound as an evaluation metric for the dialogue problem when modeling via hashing functions.

The proposed information theoretic bounds are generically applicable for any high dimensional variables. For instance, our MI LB and Entropy UB are generically applicable for other paragraph-embedding like representations [Wieting et al., 2015; Arora et al., 2016], though more efficient to compute on binary ones. See Appendix section for derivation of the bounds.

Previously, variational lower bounds on the mutual information criterion have been proposed [Barber and Agakov, 2003; Chalk et al., 2016; Gao et al., 2016; Chen et al., 2016; Alemi et al., 2017; Garg et al., 2018a] for settings where one of the two variables is fixed (say class labels). This simplifies the derivation significantly, compared to our setting.

Next, we discuss details on how to optimize kernel- or neural-hashing (Alg. 1) by maximizing our MI LB.

Hashing Func.	Hash Config.	Model	MI LB (Shuffled)	Entropy UB	NMI LB (Shuffled)	Accuracy (Baseline)
Kernel-RkNN	M=30, $\alpha=10$	Random Sel.	$-12.1 \pm 4.4$ ( $-12.1 \pm 2.4$ )	$5.1 \pm 4.6$	N.A.	$93.6 \pm 8.6$ ( $89.4 \pm 12.2$ )
		MI LB Opt.	$-4.2 \pm 1.2$ ( $-9.5 \pm 0.2$ )	<b><math>30.2 \pm 1.1</math></b>	N.A.	<b><math>75.3 \pm 16.5</math></b> ( $68.7 \pm 17.4$ )
Kernel-RkNN	M=100, $\alpha=10$	Random Sel.	$9.9 \pm 0.4$ ( $4.3 \pm 0.2$ )	$19.9 \pm 0.3$	0.50 (0.22)	$88.3 \pm 13.9$ ( $84.7 \pm 16.0$ )
		MI LB Opt.	<b><math>13.5 \pm 0.4</math></b> ( $5.6 \pm 0.3$ )	<b><math>26.1 \pm 0.3</math></b>	0.52 (0.21)	$82.7 \pm 16.2$ ( $78.4 \pm 17.3$ )
Kernel-RkNN	M=300, $\alpha=10$	Random Sel.	$15.3 \pm 0.2$ ( $8.6 \pm 0.1$ )	$22.7 \pm 0.2$	<b>0.67 (0.38)</b>	$87.6 \pm 14.7$ ( $83.9 \pm 16.6$ )
		MI LB Opt.	<b><math>15.7 \pm 0.3</math></b> ( $8.4 \pm 0.2$ )	<b><math>24.0 \pm 0.3</math></b>	0.65 (0.35)	$85.8 \pm 16.0$ ( $82.0 \pm 17.6$ )
Kernel-RMM	M=100, $\alpha=10$	Random Sel.	$10.0 \pm 0.5$ ( $3.5 \pm 0.1$ )	$23.2 \pm 0.3$	0.43 (0.15)	$85.9 \pm 11.5$ ( $79.3 \pm 13.3$ )
		MI LB Opt.	<b><math>11.4 \pm 0.5</math></b> ( $2.8 \pm 0.2$ )	<b><math>26.0 \pm 0.6</math></b>	<b>0.44 (0.11)</b>	<b><math>82.0 \pm 11.6</math></b> ( $74.5 \pm 13.2$ )
Kernel-RMM	M=300, $\alpha=10$	Random Sel.	$10.3 \pm 0.4$ ( $3.6 \pm 0.1$ )	$22.1 \pm 0.3$	0.47 (0.16)	$86.2 \pm 11.9$ ( $79.9 \pm 14.0$ )
		MI LB Opt.	<b><math>17.2 \pm 0.3</math></b> ( $8.0 \pm 0.2$ )	$23.2 \pm 0.3$	<b>0.74 (0.34)</b>	$84.6 \pm 17.2$ ( $79.9 \pm 19.9$ )

Table 2: Evaluating the quality inference of dialogue responses (hashcodes), for the depression psychotherapy dataset.

### 4.3 Details on Optimization of Kernel- or Neural-Hashing with Maximization of MI LB

In the above, we formalized an objective function, i.e. a lower bound on the mutual information criterion, to optimize all the hash functions jointly for dialogue modeling, while keeping each hash function randomized so as to preserve the property of locality sensitive hashing. In this section, we discuss optimization details specific to the kernel- or neural-hashing models as described in Sec. 3.

#### Optimizing Kernel-Hashing Models for Dialogue Modeling

For kernel-hashing in 1, we can optimize the reference set,  $\mathcal{S}^R$ , or the kernel parameters. For the selection of structures in  $\mathcal{S}^R$ , we use a greedy algorithm maximizing the proposed mutual information lower bound (in Theorem 1); see the pseudo code in Alg. 3. 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.  $\bar{\mathcal{S}} = \{S_1^p, \dots, S_N^p, S_1^t, \dots, S_N^t\}$ ; though, as noted before, the superset  $\bar{\mathcal{S}}$  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 pairs,  $\{(S_i^p, S_i^t)\}_{i=1}^N$ . 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  $\beta$  from the set  $\bar{\mathcal{S}}$ .

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 upto 300. A similar procedure can be used to optimize kernel parameters.

#### Optimizing Neural Network Architecture in Neural-Hashing Models

As mentioned previously, if using language neural models for hashing in Alg. 1, 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.

## 5 Experiments

In this section, we discuss our experimental simulations, primarily on a dataset of textual transcriptions of the audio recordings of 400 *depression therapy sessions*, with each session conducted by a human therapist with a real patient, over-all involving hundreds of patients and therapists, along with an additional dataset of interview transcripts between Larry King (host) and his guests<sup>4</sup>. For the purposes of evaluations in this paper, we put all the pairs of patient/therapist responses from different sessions together in a single set,  $\{S_i^p, S_i^t\}_{i=1}^N$ , of size  $N=42000$ . We perform random 90%-10% split of the dataset into a training (38,000 response pairs), and a test set (4200 response pairs), respectively.<sup>5</sup> Similarly, for the second dataset, we put together all pairs of host/guest responses from 75 sessions in a single set of size 8200. Further, we perform 10 trials for obtaining the statistics on evaluation metrics in each experiment, with 95% response pairs sampled from the training subset, and the same percent of pairs from the test subset.

#### Hashing Settings

The number of hashcode bits is hundred,  $H = 100$ , for both patient and therapist responses. To obtain hashcodes of responses, for each sentence in a textual response of a patient/therapist, we obtain Part of Speech Tags as additional features, known as POS.

In case of *kernel-hashing*, we use subsequence kernels [Mooney and Bunescu, 2005] for computing similarity

<sup>4</sup><http://transcripts.cnn.com/TRANSCRIPTS/1kl1.html>

<sup>5</sup>We use random seed value zero for numpy.random package.

---

Can you? They're leaving, they're leaving.  
 Yes.  
 When I took the 60, I didn't sleep for like two days.  
 I feel like it is doing nothing.  
 Talk to me and listen to me.  
 Mm-hm.  
 Uh-huh.  
 No, I'm not about to tell him. Hum-um.  
 It was one of the few things in there that I actually bought from her. None of the things I think... She was trying hard to say the right things. Where have I ever heard that from? So I leave the conversation and go, "All right, well, we'll see what happens." And what she asked in return is that when I'm angry at her, tell her. "I'm angry at you. I'm not going to talk to you for a while now." She said, "As long as I know."  
 Sure. I'll see you day after tomorrow.  
 You're not going any further in therapy right now because you haven't decided which way you want to go. It really, to me — okay, maybe I'm over playing it but it seems like a parting in the way. Which game are you going to play? Which game do you want to be good at? And you keep pussyfooting saying well, I like a little bit of this and I like a little bit of that.  
 And I can't breathe. I-I-I can still breathe, you know.  
 Right.  
 You're not going to church or nothing —  
 Yeah, it's basically the, um, the recordings are sort of a \$50 subsidy –

---

Table 3: We show first 15 responses (in raw text format) that are selected greedily for the reference set ( $S^R$ ) in *kernel-RMM hashing* model, out of 76,000 responses from the train set, with Alg. 3.

between two responses <sup>6</sup>, while similarity between a pair of words is computed using wordnet if their POS tags match. For the size of reference set  $S^R$ , we try values, 30, 100, 300 ( $\alpha = 10$ ). Reference set is initialized either randomly as a subset of all the responses in the training set (**Random Selection**), or the sub-selection is optimized with Alg. 3 (**MI LB Optimal**), with configurations  $\beta = 1000, \gamma = 100, I = 15$ . See Tab. 3. We explore two different kernel locality sensitive hashing techniques [Joly and Buisson, 2011; Garg *et al.*, 2018b], referred as **RMM** <sup>7</sup> and **RkNN** respectively, while considering the latter one as our primary choice for detailed analysis.

For the case of *neural-hashing*, we use all the responses from the training set as the reference set of size 76,000; since the reference set is very large, we use random subsets of larger size for constructing hash functions, i.e.  $\alpha = 50$ . We use an LSTM model to construct each hash function, trained using Adams algorithm [Kingma and Ba, 2014], with learning rate 1e-3, amsgrad=True, and  $l_1, l_2$  regularization coefficients of value 1e-4, and the gradients are computed with back propagation. We initialize a word in a response and its POS tag with a randomly initialized word 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 4 layers ( $L = 4, \gamma = 1000$ ), and try out different possible numbers of normal units in each layer, i.e., 4, 8, 16,

32, 64. <sup>8</sup>

### Random Forests for Inferring Therapist Hashcodes

From the above model, we obtain hashcodes for patient and therapist responses in the training and test subsets, serving as ground truth for the task of inferring the hashcode of the therapist response, given the hashcode of a patient response. For inferring each bit of therapist code, we train an individual Random Forest (RF) classifier containing 100 decision trees. All of the hundred RF classifiers (since  $H = 100$ ) share the same features (i.e. the hashcodes of patient responses) though trained independent of each other.

## 5.1 Evaluation Results

**Evaluation metrics** Our primary evaluation metric is our proposed MI lower bound (**MI LB**) in Theorem 1. We also use the proposed upper bound on the entropy of therapist hashcodes in Theorem 2 to evaluate the quality of hashcodes as representations of responses (**Entropy UB**). In addition, we express the ratio of MI LB and Entropy UB as *Normalized MI LB* (NMI LB). It is also an interesting to analyze MI LB for random pairings between patient and therapist responses (denoted as "Shuffled"); same applies for NMI LB.

Interpreting the accuracy metric (**Accuracy**), i.e. the accuracy of inferring each hashcode bit using the RF classifiers, requires a more detailed discussion. Since the class labels (therapist hashcodes) are not fixed, the accuracy is inversely correlated with the Entropy UB metric. We also obtain the baseline accuracy (**Baseline**), using a trivial classifier that always chooses the most-frequent class label. While analyzing the absolute numbers of accuracy is not meaningful here, we can see the relative improvement of accuracy w.r.t. the accuracy of the baseline dummy classifier. 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 computed over

<sup>6</sup> $\lambda = 0.8$ , subsequence of length up to 8 are matched

<sup>7</sup>In this hashing technique, an SVM is learned for each hash function. We use  $C = 0.1$  in SVM, keeping it highly regularized.

<sup>8</sup>We keep the number of units small, since  $\alpha$  is small.



---

Run away and, and be by myself.

**Um-hum.**

(i) **Uh-huh.** (ii) **Oh.** (iii) **That doesn't pan out.**

---

Yeah, we'll be in touch about all that stuff.

**Okay. So very nice to see you again.**

(i) **OK, I see.** (ii) **Kind of like that-** (iii) **Good, good. What's new anything ?**

---

Sensitive. Yeah. (inaudible) just like (inaudible) very uncomfortable.

**Uh-huh.**

(i) **What's going on with the moods?** (ii) **Big people don't feel this way like?** (iii) **Scares you how?**

---

Yeah. It just feels moderately depressing. My head spins

**Do you feel depressed?**

(i) **So eventually it will die out.** (ii) **So it may peter out later.** (iii) **We can stop for now.**

---

Well, am I kidding myself when I'm saying nothing's bothering me? So that bothers me.

**When you don't - whether you can really trust your own feelings.**

(i) **To be at - all of us.** (ii) **I'm sure.** (iii) **That sounds exhausting, yeah.**

---

(laughs) Um, yeah. (laughs)

**Forever.**

(i) **OK, I see.** (ii) **Okay, I understand.** (iii) **All right, take it.**

---

But it's like that's what I've been doing. And I don't, I can't scrape up the money to go back to school.

**Yeah.**

(i) **Um-hum.** (ii) **Mm-hmm.** (iii) **Hmmm.**

---

Table 4: Here, we present some examples of textual response choices produced with our hashing based approach (raw text format). For a patient response (Black), we show the ground truth response from a human therapist (Blue), and three textual response choices generated with kernel-RMM model (Magenta) via mapping of the inferred therapist hashcode to hashcode-indexed therapist responses from the training set.

all (100) hashcode bits and over 10 trials. For each of the metrics, *higher values* mean better performance. In addition, by subtracting MI LB from Entropy UB, we obtain the inference quality.

### Depression Dataset Results

We present the results for our primary evaluation on the depression dataset in Tab. 2. *Overall, our MI LB optimization based approach is outperforming the alternatives in most of the settings, for the MI LB as well as Entropy UB metrics (best results shown in boldface).* We will first discuss in more detail our results for kernel-based hashing.

**Kernel-Hashing** Since the performance of a kernel-hashing model varies for a random initialization of  $S^R$ , we perform multiple trials in that case, and report numbers for the trial which gives the median value for MI LB. For *RkNN-Kernel* hashing, we vary sizes of the reference set  $S^R$ , i.e.  $M \in \{30, 100, 300\}$ . For  $M = 30$ , MI LB (-11.5) and Entropy UB (5.7) values are low when selecting the reference set elements randomly (*Random Selection*). On the other hand, if the reference set is optimized (*MI LB Optimal*), MI LB (4.4) and Entropy UB (30.1) are of much higher values, as desired. For  $M = 100$ , MI LB values increase significantly compared to the values for  $M = 30$ ; thus high value of  $M$  is advantageous. We see the benefits of optimizing the reference set (MI LB: 13.6, Entropy UB: 26.2), compared to the Random Selection case (MI LB: 9.8, Entropy UB: 19.9), for  $M = 100$  too. For a further increase in the size of the reference set,  $M = 300$ , we obtain even higher values for MI LB (15.7), though not a significant increase in Entropy UB (24.0), especially when selecting the reference set randomly. In this setting, the advantage of optimizing the reference set is somewhat marginal. For robustness, it is always good to optimize

the set rather than a random selection.

For *kernel-RMM*, we use  $M = 100$ , obtaining similar results as with kernel-RkNN hashing. Here it is interesting to see that the ratio of MI LB (11.4) w.r.t. its baseline of MI LB for shuffled pairs (2.8), is significantly higher (4.1) than other models. This ratio should serve as another metric derived from MI LBs, to judge the quality of a dialogue model (higher value of the ratio is better).

With an increase in  $M$  when selecting the reference set randomly, we observe decrease in baseline accuracy since the Entropy UB increases in value. Also, when optimizing the reference set instead of Random selection, Entropy UB increases, and therefore the baseline accuracy decreases. Overall, when the difference between Entropy UB and MI LB (quantifying inference quality) is high, the positive gap between the accuracy numbers and the baseline accuracy numbers is more significant. Variance for accuracy is high because for some bits in hashcodes, the distribution of 0s and 1s is imbalanced (say, 99%-1%), making the inference very challenging for those bits.

Herein, it is also interesting to observe MI LB values for the “Shuffled” setting (presented in brackets), i.e. computing MI LB for random pairings of patient/therapist responses. This should serve as a good metric to analyze a hashing based dialogue model as well. Since we are modeling responses from all therapy sessions put together into a single set, one should expect that a good model should have low value of MI LB for the shuffled setting; though, it is natural to expect a minimal value of mutual information even between randomly paired responses of patient & therapist, from different sessions, due to the common theme of depression psychotherapy across the sessions.

Hashing Func.	Hash Config.	Model	MI LB (Shuffled)	Entropy UB	NMI LB (Shuffled)	Accuracy (Baseline)
Kernel-RkNN	M=100, $\alpha$ =10	Random Sel.	15.1 $\pm$ 0.5 (7.3 $\pm$ 0.2)	22.6 $\pm$ 0.4	0.67 (0.32)	87.7 $\pm$ 12.9 (81.8 $\pm$ 17.3)
Kernel-RkNN	M=100, $\alpha$ =10	MI LB Opt.	16.8 $\pm$ 0.6 (7.9 $\pm$ 0.4)	25.3 $\pm$ 0.6	0.66 (0.31)	86.2 $\pm$ 13.8 (79.5 $\pm$ 17.2)
Kernel-RkNN	M=300, $\alpha$ =10	Random Sel.	<b>20.4 <math>\pm</math> 0.4</b> (11.7 $\pm$ 0.2)	26.9 $\pm$ 0.5	<b>0.76 (0.43)</b>	88.3 $\pm$ 11.0 (80.7 $\pm$ 15.2)
Neural-RLSTM	M=18000, $\alpha$ =50	MI LB Opt.	<b>49.5 <math>\pm</math> 0.3</b> (28.0 $\pm$ 0.4)	<b>60.6 <math>\pm</math> 0.2</b>	<b>0.82 (0.46)</b>	<b>69.1 <math>\pm</math> 7.8</b> (55.8 $\pm$ 8.3)

Table 5: Evaluating the quality inference of dialogue responses (hashcodes), for the Larry King dataset.

## Neural-Hashing

For the optimization of Neural-RLSTM hashing model, Alg. 2 gives a simple LSTM architecture of [16, 64, 16, 8], i.e. four layers, with 16 LSTM units in the first layer (bottom layer receiving the inputs), 64 LSTM units in the second layer, and so on. For this model, we obtain relatively higher values for MI LB (17.2), compared to the kernel-hashing above. Besides the optimized architecture for LSTM models, we also tried manually built architectures; for instance, if using simple LSTM model of single layer with 10 units for each hash function, we obtain MI LB value, 6.7, that is significantly lower compared to the optimized Neural-RLSTM model (17.2), and many of the kernel-hashing models above.

The above results demonstrate that our approach of optimizing hashing functions by maximizing our novel MI LB leads to higher values for the evaluation metrics as desired, across the kernel- as well as neural-hashing models. There is a trade off between kernel-hashing and neural-hashing. In kernel-hashing, as per the selected responses in the optimized reference set, one can get insights on what patterns are relevant for a dialog; for e.g., see Tab. 3. On the other hand with neural-hashing, one can take advantage of the vast variety of neural language models available.

Further, we can map an inferred hashcode to textual responses using a repository of indexed sentences. See Tab. 4.

## Larry King Dataset Results

Some selected results for the Larry King dataset are presented in Tab. 5. Herein, it is interesting to see that we get very high value of MI LB, and Entropy UB with our neural hashing approach (optimized architecture is [8,32]), in comparison to our kernel hashing. Also, mean accuracy with neural hashing is significantly higher than the baseline accuracy number. Though, NMI LB values for the two hashing approaches are relatively close.

## 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. Moreover, in order to characterize the general quality of hashcodes as response representations, we propose a tight upper bound on the joint entropy of hashcodes. We performed empirical evaluation of the proposed approach on the dataset containing

depression therapy sessions between real patients and therapists. We optimized locality sensitive hashing models, based on kernel functions or neural language models, by maximizing the proposed MI lower bound as an objective function. Our results consistently demonstrate superior performance of the proposed approach over several alternatives, as measured by several evaluation metrics.

## References

- [Alemi *et al.*, 2017] Alex Alemi, Ian Fischer, Josh Dillon, and Kevin Murphy. Deep variational information bottleneck. 2017.
- [Althoff *et al.*, 2016] Tim Althoff, Kevin Clark, and Jure Leskovec. Large-scale analysis of counseling conversations: An application of natural language processing to mental health. *Transactions of the Association for Computational Linguistics*, 4:463, 2016.
- [Arora *et al.*, 2016] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. 2016.
- [Asghar *et al.*, 2017] Nabiha Asghar, Pascal Poupart, Xin Jiang, and Hang Li. Deep active learning for dialogue generation. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (\* SEM 2017)*, pages 78–83, 2017.
- [Barber and Agakov, 2003] David Barber and Felix Agakov. 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, 2003.
- [Bartl and Spanakis, 2017] Alexander Bartl and Gerasimos Spanakis. A retrieval-based dialogue system utilizing utterance and context embeddings. *arXiv preprint arXiv:1710.05780*, 2017.
- [Chalk *et al.*, 2016] Matthew Chalk, Olivier Marre, and Gasper Tkacik. Relevant sparse codes with variational information bottleneck. In *Advances in Neural Information Processing Systems*, pages 1957–1965, 2016.
- [Chen *et al.*, 2016] Xi Chen, Yan Duan, Rein Houthoofd, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180, 2016.
- [Collins and Duffy, 2002] Michael Collins and Nigel Duffy. Convolution kernels for natural language. In *Advances*

- in *neural information processing systems*, pages 625–632, 2002.
- [Fitzpatrick *et al.*, 2017] Kathleen Kara Fitzpatrick, Alison Darcy, and Molly Vierhile. 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, 2017.
- [Gao *et al.*, 2016] Shuyang Gao, Greg Ver Steeg, and Aram Galstyan. Variational information maximization for feature selection. In *Advances in Neural Information Processing Systems*, pages 487–495, 2016.
- [Garg *et al.*, 2018a] Sahil Garg, Aram Galstyan, Greg Ver Steeg, Irina Rish, Guillermo Cecchi, and Shuyang Gao. Efficient representation for natural language processing via kernelized hashcodes. *arXiv preprint arXiv:1711.04044*, 2018.
- [Garg *et al.*, 2018b] Sahil Garg, Greg Ver Steeg, and Aram Galstyan. Stochastic learning of nonstationary kernels for natural language modeling. *arXiv preprint arXiv:1801.03911*, 2018.
- [Haussler, 1999] David Haussler. Convolution kernels on discrete structures. Technical report, 1999.
- [He *et al.*, 2017] He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. *arXiv preprint arXiv:1704.07130*, 2017.
- [Joly and Buisson, 2011] Alexis Joly and Olivier Buisson. Random maximum margin hashing. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 873–880. IEEE, 2011.
- [Jurafsky and Martin, 2014] D Jurafsky and JH Martin. Dialogue systems and chatbots. *Speech and language processing*, 3, 2014.
- [Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [Komorowski and Trzcinski, 2017] Michal Komorowski and Tomasz Trzcinski. Random binary trees for approximate nearest neighbour search in binary space. *arXiv preprint arXiv:1708.02976*, 2017.
- [Kosovan *et al.*, ] Sofiia Kosovan, Jens Lehmann, and Asja Fischer. Dialogue response generation using neural networks with attention and background knowledge.
- [Kraskov *et al.*, 2004] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 69:066138, 2004.
- [Kraskov *et al.*, 2005] Alexander Kraskov, Harald Stögbauer, Ralph G Andrzejak, and Peter Grassberger. Hierarchical clustering using mutual information. *EPL (Europhysics Letters)*, 70(2):278, 2005.
- [Kulis and Grauman, 2009] Brian Kulis and Kristen Grauman. Kernelized locality-sensitive hashing for scalable image search. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2130–2137. IEEE, 2009.
- [Kulis and Grauman, 2012] Brian Kulis and Kristen Grauman. Kernelized locality-sensitive hashing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6):1092–1104, 2012.
- [Lewinsohn *et al.*, 1990] Peter M Lewinsohn, Gregory N Clarke, Hyman Hops, and Judy Andrews. Cognitive-behavioral treatment for depressed adolescents. *Behavior Therapy*, 21(4):385–401, 1990.
- [Lewis *et al.*, 2017] Mike Lewis, Denis Yarats, Yann N Dauphin, Devi Parikh, and Dhruv Batra. Deal or no deal? end-to-end learning for negotiation dialogues. *arXiv preprint arXiv:1706.05125*, 2017.
- [Li and Jurafsky, 2016] Jiwei Li and Dan Jurafsky. Neural net models for open-domain discourse coherence. *arXiv preprint arXiv:1606.01545*, 2016.
- [Li *et al.*, 2015] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*, 2015.
- [Li *et al.*, 2017] Jiwei Li, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*, 2017.
- [Liu *et al.*, 2016] Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*, 2016.
- [Lowe *et al.*, 2017] Ryan Lowe, Michael Noseworthy, Iulian V Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. Towards an automatic turing test: Learning to evaluate dialogue responses. *arXiv preprint arXiv:1708.07149*, 2017.
- [Mooney and Bunesco, 2005] Raymond J Mooney and Razvan C Bunesco. Subsequence kernels for relation extraction. In *Proc. of NIPS*, pages 171–178, 2005.
- [Norouzi *et al.*, 2014] Mohammad Norouzi, Ali Punjani, and David J Fleet. Fast exact search in hamming space with multi-index hashing. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1107–1119, 2014.
- [Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, 2002.
- [Serban *et al.*, 2015] Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. A survey of available corpora for building data-driven dialogue systems. *arXiv preprint arXiv:1512.05742*, 2015.
- [Serban *et al.*, 2016] Iulian Vlad Serban, Ryan Lowe, Laurent Charlin, and Joelle Pineau. Generative deep neural networks for dialogue: A short review. *arXiv preprint arXiv:1611.06216*, 2016.

- [Serban *et al.*, 2017a] Iulian Vlad Serban, Tim Klinger, Gerald Tesauro, Kartik Talamadupula, Bowen Zhou, Yoshua Bengio, and Aaron C Courville. Multiresolution recurrent neural networks: An application to dialogue response generation. In *AAAI*, pages 3288–3294, 2017.
- [Serban *et al.*, 2017b] Iulian Vlad Serban, Alessandro Sordani, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, pages 3295–3301, 2017.
- [Shao *et al.*, 2017] Yuanlong Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 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, 2017.
- [Studený and Vejnárová, 1998] Milan Studený and Jirina Vejnárová. The multiinformation function as a tool for measuring stochastic dependence. In *Learning in graphical models*, pages 261–297. Springer, 1998.
- [Ver Steeg and Galstyan, 2014] Greg Ver Steeg and Aram Galstyan. Discovering structure in high-dimensional data through correlation explanation. In *Advances in Neural Information Processing Systems 27*. 2014.
- [Wang *et al.*, 2014] Jingdong Wang, Heng Tao Shen, Jingkuan Song, and Jianqiu Ji. Hashing for similarity search: A survey. *arXiv preprint arXiv:1408.2927*, 2014.
- [Wang *et al.*, 2017] Jingdong Wang, Ting Zhang, Nicu Sebe, Heng Tao Shen, et al. A survey on learning to hash. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
- [Watanabe, 1960] Satoshi Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of research and development*, 4(1):66–82, 1960.
- [Wen *et al.*, 2016] Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*, 2016.
- [Wieting *et al.*, 2015] John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*, 2015.
- [Wu *et al.*, 2017] Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. 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, 2017.
- [Zhai and Williams, 2014] Ke Zhai and Jason D Williams. Discovering latent structure in task-oriented dialogues. In *ACL (1)*, pages 36–46, 2014.

## A Derivations of the Information Theoretic Bounds

Before the discussion of our novel lower bound, we introduce the information-theoretic quantity called *Total Correlation* ( $\mathcal{TC}$ ), which captures non-linear correlation among the dimensions of a random variable  $\mathcal{X}$ , i.e.,

$$\begin{aligned}\mathcal{TC}(\mathcal{X}; M_h) &= \sum_j \mathcal{H}(\mathcal{X}(j); M_h) - \mathcal{H}(\mathcal{X}; M_h); \\ \mathcal{TC}(\mathcal{X} : \mathcal{Y}; M_h) &= \mathcal{TC}(\mathcal{X}; M_h) - \mathcal{TC}(\mathcal{X}|\mathcal{Y}; M_h).\end{aligned}\quad (5)$$

Intuitively, (5) describes the amount of information within  $\mathcal{X}$  that can be explained by  $\mathcal{Y}$ .

Along these lines, the mutual information quantity between the hashcodes can be decomposed as in Lemma 1 below.

**Lemma 1** (Mutual Information Decomposition). *Mutual Information between  $\mathcal{C}_t$  and  $\mathcal{C}_p$  is decomposed as follows:*

$$\begin{aligned}\mathcal{I}(\mathcal{C}_t : \mathcal{C}_p; M_h) \\ = \sum_j \mathcal{I}(\mathcal{C}_t(j) : \mathcal{C}_p; M_h) - \mathcal{TC}(\mathcal{C}_t : \mathcal{C}_p; M_h).\end{aligned}\quad (6)$$

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 [Gao *et al.*, 2016], as in Lemma 2 below.

**Lemma 2.** *Marginal mutual information for each bit in therapist hashcodes,  $\mathcal{I}(\mathcal{C}_t(j) : \mathcal{C}_p; M_h)$ , is lower bounded as,*

$$\begin{aligned}\mathcal{I}(\mathcal{C}_t(j) : \mathcal{C}_p; M_h) \geq \\ \mathcal{H}(\mathcal{C}_t(j); M_h) - \langle \log q(\mathcal{C}_t(j)|\mathcal{C}_p; M_h) \rangle_{p(\mathcal{C}_t(j), \mathcal{C}_p; M_h)}.\end{aligned}\quad (7)$$

Herein,  $\mathcal{H}(\mathcal{C}_t(j); M_h)$  is easy to compute because  $\mathcal{C}_t(j)$  is one-dimensional. For each of the proposal distributions  $q(\mathcal{C}_t(j)|\mathcal{C}_p; M_h)$ , we propose to use a Random Forest (RF) classifier [Garg *et al.*, 2018a].

In reference to the second term of RHS in (6), it is computationally intractable to compute the total correlation expression  $\mathcal{TC}(\mathcal{C}_t : \mathcal{C}_p; M_h)$ , which denotes the total correlations between bits of  $\mathcal{C}_t$ , explainable by  $\mathcal{C}_p$ . So, we would also like to obtain an upper bound of  $\mathcal{TC}(\mathcal{C}_t : \mathcal{C}_p; M_h)$ , 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.**  *$\mathcal{TC}(\mathcal{C}_t : \mathcal{C}_p; M_h)$  can be upper bounded as:*

$$\mathcal{TC}(\mathcal{C}_t : \mathcal{C}_p; M_h) \leq \mathcal{TC}(\mathcal{C}_t : \mathcal{Y}^*; M_h); \quad (8)$$

wherein  $|\cdot|$  denotes the dimensionality of a random variable.

Although it is intractable to compute the original term  $\mathcal{TC}(\mathcal{C}_t : \mathcal{C}_p; M_h)$ , it is possible to compute  $\mathcal{TC}(\mathcal{C}_t : \mathcal{Y}^*; M_h)$  for a latent variable representation  $\mathcal{Y}^*$  of  $\mathcal{C}_t$  that maximally explains the Total Correlations in  $\mathcal{C}_t$ .

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  $\mathcal{Y}^*$  [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  $\mathcal{TC}(\mathcal{C}_t : \mathcal{Y}; M_h)$  above, the optimized quantity may not be an upper bound of  $\mathcal{TC}(\mathcal{C}_t : \mathcal{C}_p; M_h)$ , but rather an approximation. Also, the upper bound would not be tight if  $\mathcal{C}_p$  doesn't explain much of total correlations in  $\mathcal{C}_t$ . Further, for even more computation cost reductions during the learning, the dimension of the latent representation  $\mathcal{Y}$  can be kept much smaller than the dimension of hashcodes, i.e.  $|\mathcal{Y}| \ll |\mathcal{C}_p|$  for  $|\mathcal{C}_p| \gg 1$ ; this is because even a small number of latent variables should explain most of the total correlations for practical purposes as demonstrated in [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 (3) in Theorem 1.

Along same lines, we derive the tight upper bound on joint entropy of hashcodes in Theorem 2. From the definition of Total Correlation above (5), we have the following,

$$\begin{aligned} \sum_j \mathcal{H}(\mathcal{C}_t(j); M_h) - \mathcal{TC}(\mathcal{C}_t; M_h) &= \mathcal{H}(\mathcal{C}_t; M_h), \\ \mathcal{TC}(\mathcal{C}_t; M_h) &= \mathcal{TC}(\mathcal{C}_t : \mathcal{Y}^*; M_h) + \mathcal{TC}(\mathcal{C}_t | \mathcal{Y}^*; M_h), \end{aligned}$$

and finally the expression below.

$$\begin{aligned} \sum_j \mathcal{H}(\mathcal{C}_t(j); M_h) - \mathcal{TC}(\mathcal{C}_t : \mathcal{Y}^*; M_h) \\ = \mathcal{H}(\mathcal{C}_t; M_h) + \mathcal{TC}(\mathcal{C}_t | \mathcal{Y}^*; M_h) \end{aligned}$$

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.*, 2018a]. Their lower bounds works only when one of the variables is fixed, say if  $\mathcal{C}_t$  were fixed. In our objective, not only  $\mathcal{C}_t$  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  $\mathcal{H}(\mathcal{C}_t; M_h)$  as well, which should be hard to obtain, we can not use the above mentioned variational lower bounds for our problem as such. Besides, it is also non-trivial to find an appropriate proposal distribution  $q(\mathcal{C}_t | \mathcal{C}_p; M_h)$ . Therefore, we adopt a different approach for obtaining a novel lower bound on the mutual information quantity, as described above.