Mix-and-Match: Scalable Dialog Response Retrieval using Gaussian Mixture Embeddings

Gaurav Pandey
IBM Research AI
New Delhi
gpandey1@in.ibm.com

Danish Contractor
IBM Research AI
New York
danish.contractor@ibm.com

Sachindra Joshi
IBM Research AI
New Delhi
jsachind@in.ibm.com

Abstract

Embedding-based approaches for dialog response retrieval embed the context-response pairs as points in the embedding space. These approaches are scalable, but fail to account for the complex, many-to-many relationships that exist between context-response pairs. On the other end of the spectrum, there are approaches that feed the context-response pairs jointly through multiple layers of neural networks. These approaches can model the complex relationships between context-response pairs, but fail to scale when the set of responses is moderately large (>1000). In this paper, we propose a scalable model that can learn complex relationships between context-response pairs. Specifically, the model maps the contexts as well as responses to probability distributions over the embedding space. We train the models by optimizing the Kullback-Leibler divergence between the distributions induced by context-response pairs in the training data. We show that the resultant model achieves better performance as compared to other embedding-based approaches on publicly available conversation data.

1 Introduction

Retrieval-based response predictors (Ji et al., 2014; Yan et al., 2016; Wu et al., 2017a; Bartl and Spanakis, 2017; Whang et al., 2021; Xu et al., 2021; Han et al., 2021; Su et al., 2021; Gu et al., 2020) retrieve the response from a predefined set of responses given the dialog context. Such methods find application in a variety of real-world dialog modeling and collaborative human-agent tasks. For instance, dialog modeling frameworks typically utilize the notion of "intents" and "dialog flows" which aim to model the "goal" of a user-utterance (Aronsson et al., 2021). To make task of building and identifying such intents easier, some tools mine conversation logs to identify responses that are often associated with dialog contexts (intents) (Dhoolia et al., 2021) and then surface these responses for review by humans. These reviewed responses are then modeled into the dialog flow for different intents. Another instance, of human-agent collaboration powered by system returned responses is in 'Agent Assist' environments where a system makes recommendations to a customer-support or contact-center agent in real-time(Fadnis et al., 2020).

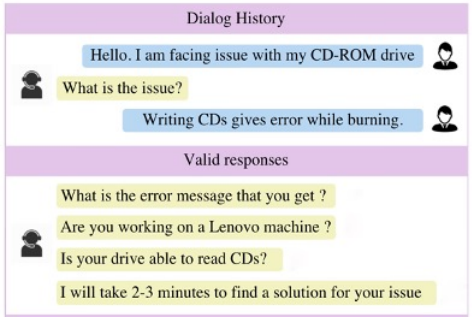


Figure 1: An example of a context with multiple valid responses. Note that each response contains different information and hence must have embeddings that are far way from each other. However, embedding-based approaches for retrieval attempt to bring all such responses close to the context and hence, close to each other.

The success of a good response retrieval system lies in learning a good similarity function between the context and the response. In addition, it also needs to be scalable so that it can retrieve responses from the universe of responses efficiently. These two requirements present a trade-off between the richness of scoring and scalability, as discussed below.

Trade-off between Scoring and Scalability: Typically, in neural dialog retrieval models, the contexts and the responses in the conversation logs are embedded as points in the embedding space (Lowe et al., 2015). Approaches such as contrastive learning (Bromley et al., 1993) are then used to ensure that the context is closer to the ground-truth response than the other responses. Figure 1 shows

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 3273 - 3287
December 7-11, 2022 ©2022 Association for Computational Linguistics

a dialog context followed by multiple responses. Despite the apparent diversity among responses, all the responses are valid for the dialog context. Similarly, a generic response may be a valid response for several dialog contexts. A typical embedding-based approach for retrieval would bring the embedding of the dialog context close to the embedding of all the valid responses (Karpukhin et al., 2020; Yu et al., 2021; Xiong et al., 2021; Luan et al., 2021a). However this has the undesirable effect of making the valid, but diverse, responses gravitate towards each other in the embedding space.

Thus, typical embedding-based approaches for retrieval fail to capture the complex, many-to-many relationships that exist in conversations. More complex matching networks such as Sequential Matching Networks (Wu et al., 2017a) and BERT (Chen et al., 2021b) based cross-encoders jointly feed the context-response pairs through multiple layers of neural networks for generating the similarity score. While these approaches have proven to be effective for response retrieval, they are very expensive in terms of inference time. Specifically, if N_c is the total number of dialog contexts and N_r is the total number of responses available for retrieval during inference, these methods have a time complexity of $O(N_c N_r)$. Hence, they can't be used in a real-world setting for retrieving from thousands of responses.

Contributions: In this paper we present a scalable and efficient dialog-retrieval system that maps the contexts as well as the responses to probability distributions over the embedding space (instead of points in the embedding space). To capture the complex many-to-many relationships between the context and response, we use multimodal distributions such as Gaussian mixtures to model each context and response. The resultant model is referred to as 'Mix-and-Match'.

Intuitively, if a response is a valid response for a given dialog context, we want the corresponding probability distribution to be "close" to the context distribution. We formalize this notion of closeness among distributions by using Kullback-Leibler (KL) divergence. Specifically, we minimize the Kullback-Leibler divergence between the context distribution and the distribution of the ground-truth response while maximizing the divergence from the distributions of other negatively-samples responses. We derive approximate but closed-form expressions for the KL divergence when the underlying distributions are Gaussian mixtures. This approximation significantly alleviates the computation cost of KL-divergence, thereby making it suitable for use in real-world settings. We demonstrate our work on two publicly available dialog datasets – Ubuntu Dialog Corpus (v2)(Lowe et al., 2015) and the Twitter Customer Support dataset[1] as well as on an internal real-world technical support dataset. Using automated as well as human studies, we demonstrate that Mix-and-Match outperforms recent embedding-based retrieval methods. Due to space limitations, we discuss a few related works in the Appendix.

2 Mix-and-Match

We consider a dialog to be a sequence of utterances $(u_1, \ldots, u_n)$. At any time-step t, the set of utterances prior to that time-step is referred to as the context. The utterance that immediately follows the context[2] is referred to as the response. Instead of modeling the context and response as point embeddings, we use probability distributions induced by the context and the response on the embedding space, denoted as $p_c(z)$ and $p_r(z)$[3] respectively, where z is any point in the embedding space $\mathbb{R}^d$.

2.1 Overview

An overview of the model is shown in Figure 2. The context and response are first encoded using a pre-trained BERT model. The model consists of a Gaussian Mixture Parameter Generator, $\pi(X, K)$, which takes as input an encoded text sequence X along with the number of Gaussian Mixtures, K and then returns the means μ_k and variance σ_k^2 for the every Gaussian mixture component $k \in \{1, \ldots K\}$, as its output. The encoded representations of the context and response from BERT are used to generate Gaussian Mixture distributions over the embedding space $\mathbb{R}^d$ using the parameter generator π. We then compute the KL divergence between the context and response distributions and use contrastive loss to bring the context closer to the ground-truth response as compared to other, negatively-sampled responses.

[1] https://www.kaggle.com/thoughtvector/customer-support-on-twitter

[2] We use the words context and dialog context interchangeably throughout the paper.

[3] Formally, these are densities induced by the corresponding distributions

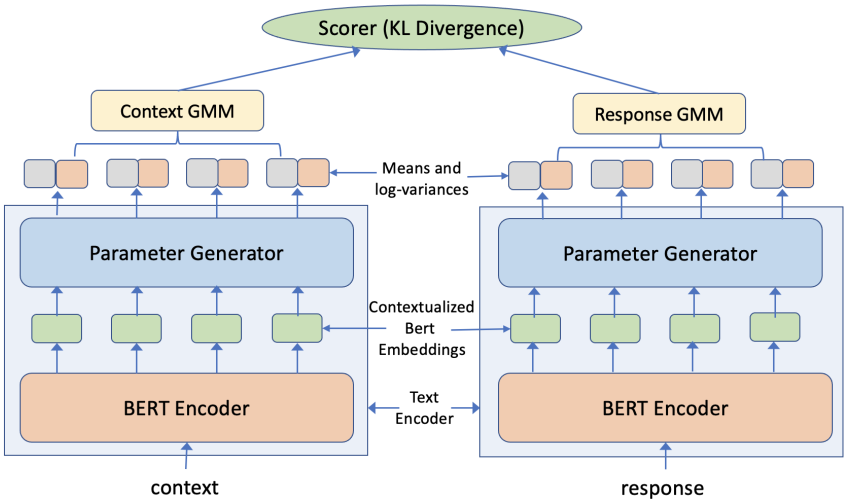


Figure 2: An overview of our model - Mix-and-Match.

2.2 Text Encoder

The text encoder maps the raw text to a contextualized embedding. Given a text sequence, we split it into tokens using the BERT tokenizer (Devlin et al., 2019). The BERT encoder (Devlin et al., 2019) takes the tokens as input and outputs the contextualized embedding of each token at the output. These embeddings are denoted as X $(x_1, \ldots, x_m)$, where m is the number of tokens in the text sequence.

2.3 Parameter Generation of Gaussian Mixtures

We use the parameter generator π with the inputs X and K to generate the parameters $\mu_k(X), \sigma_k^2(X)$ for each component of the mixture $k \in \{1, \ldots K\}$. For simplicity, we assume a restricted form of Gaussian mixture that assigns equal probability to each Gaussian component. Further, we also assume that Gaussian components are axis-aligned that is, their covariance matrix is diagonal. Specifically, the probability distribution over the embedding space $\mathbb{R}^d$ induced by the input text embeddings X is as follows:

$$p_X(z) = \frac{1}{K} \sum_{k=1}^{K} \mathcal{N}(z; \mu_k(X), \sigma_k^2(X)) \quad (1)$$

Given an input sequence of text X with token embedding representations $x_1 \ldots x_{|X|}$, we initialize K trainable embeddings $e_1, \ldots, e_K$ with same dimensions as x_i. These trainable embeddings are used to attend on X to get attended token representations $a_1, \ldots, a_K$. That is, $a_k = \sum_{i=1}^{m} \alpha_{ik} x_i$, where α_{ik} are the normalized attention weights and are defined as follows:

$$\alpha_{ik} = \frac{\exp(x_i^{\mathrm{T}} e_k)}{\sum_{\tilde{i}=1}^{m} \exp(x_{\tilde{i}}^{\mathrm{T}} e_k)}, 1 \leq k \leq K \quad (2)$$

Finally, the attended token embeddings are passed through two linear maps in parallel to generate the mean and log-variance of each Gaussian component in the mixture. That is, $\mu_k = f_1(a_k)$ and $\log(\sigma_k^2)) = f_2(a_k)$, where f_1 and f_2 are linear maps.

2.4 Context and Response Encodings

Given the dialog context c and response r, we generate the Gaussian Mixture representations $p_c(z)$ (for context) and $p_r(z)$ (for response) using π, with K and L components respectively. The Gaussian components of the mixture are denoted as $p_c(z; k)$ (for context) and $p_r(z; \ell)$ (for response) and are given by

$$p_c(z; k) = \mathcal{N}(z; \mu_k(\mathbf{c}), \sigma_k^2(\mathbf{c})) \quad (3)$$

$$p_r(z; \ell) = \mathcal{N}(z; \mu_\ell(\mathbf{r}), \sigma_\ell^2(\mathbf{r})) \quad (4)$$

where $\mu_k(\mathbf{c})$ and $\sigma_k^2(\mathbf{c})$ are the means and variances of the k^{th} Gaussian component for the context, and $\mu_\ell(\mathbf{r})$ and $\sigma_\ell(\mathbf{r})$ are the means and variances of the ℓ^{th} Gaussian component of the response. The parameters of the text encoders (BERT and π module) for context and response are not shared.

2.5 Scoring Function

We want the context distribution to be 'close' to the distribution of the ground-truth response while

simultaneously being away from distributions induced by other responses. We use the KL divergence to quantify this degree of closeness. The KL divergence between the distributions p_r and p_c over the embedding space $\mathbb{R}^d$ is given by

$$\mathrm{KL}(p_r||p_c) = \int_{z\in\mathbb{R}^d} p_r(z)\log\frac{p_r(z)}{p_c(z)}\mathrm{d}z \tag{5}$$

This integral has a closed form expression if both p_r and p_c are Gaussian. However, for Gaussian mixtures, this integral needs to be approximated. We derive the following approximation to the KL divergence between two GMMs.

Theorem 1. *Let p_r and p_c be two Gaussian mixture distributions with L and K Gaussian components respectively as defined in (3) and (4) respectively. The* KL *divergence between the two GMMs can be approximated by the following quantity*

$$\begin{aligned}\mathrm{KL}(p_r||p_c) &\approx \log(K/L)+ \\ &\frac{1}{L}\sum_{\ell=1}^{L}\min_{k\in\{1,\ldots,K\}}\mathrm{KL}(p_r(.;\ell)||p_c(.;k)),\end{aligned} \tag{6}$$

where $p_c(.;k)$ and $p_r(.;\ell)$ are the k^{th} and ℓ^{th} Gaussian component of the context and response distributions as defined in (3) and (4).

A detailed derivation of the above approximation is provided in the Appendix. Note that the theorem above holds even when the individual components of the mixture are not Gaussian.

When the components are Gaussian, the KL divergence between the components can be tractably computed using the following equation:

$$\begin{aligned}&\mathrm{KL}(p_r(.;\ell)||p_c(.;k)) = -d/2+ \\ &\frac{1}{2}\sum_{j=1}^{d}\left[\log\frac{\sigma_{kj}^2(\mathbf{c})}{\sigma_{\ell j}^2(\mathbf{r})}+\frac{\sigma_{\ell j}^2(\mathbf{r})+(\mu_{\ell j}(\mathbf{r})-\mu_{kj}(\mathbf{c}))^2}{\sigma_{kj}(\mathbf{c})^2}\right],\end{aligned} \tag{7}$$

where d is the dimension of the embedding space. Using equations (6) and (7), we get a closed form approximation to the Kullback-Leibler divergence between context and response GMMs.

2.6 Loss Function

We use N-pair contrastive loss (Sohn, 2016) for training the distributions induced by the context and response. Intuitively, given a batch $\mathcal{B}$ of context-response pairs, we minimize the KL divergence between the context and the true response while simultaneously maximizing the KL divergence with respect to other randomly selected responses. The loss for a given context-response pair $(\mathbf{c},\mathbf{r})$ can be written as

$$\text{loss} = -\log\frac{\exp(-\mathrm{KL}(p_r||p_c))}{\sum_{\bar{r}\in\mathcal{B}}\exp(-\mathrm{KL}(p_{\bar{r}}||p_c))} \tag{8}$$

We average this loss across all the context-response pairs in the batch and minimize it during training. The BERT encoders, the randomly initialized embeddings as well as the linear layers for computing the means and variances, are trained in an end-to-end manner.

2.7 Inference

During inference, we are provided a context and a collection of responses to select from. We map the context as well as the list of responses to their corresponding probability distributions over the embedding space. Next, we compute the KL divergence between the distribution induced by the context and every response in the list. Using the equation derived in (6), this can be computed efficiently and involves standard matrix operations only. We select the top-m responses that have the least KL divergence, where m is specified during evaluation.

3 Related Work

Our work is broadly related with two current areas of research - response retrieval and probabilistic embeddings.

3.1 Response Retrieval Systems

Depending on how the context and responses are encoded for retrieval, response-retrieval approaches can be classified into methods that use: (i) independent encodings (ii) joint encodings.

Independent Encodings: In these methods, the contexts and the responses are encoded independently and the resultant embeddings are fed to a scoring function. A common architecture employed by neural methods for dialog retrieval is a dual encoder. Here, the context and responses are encoded using a shared architecture but in different parameter spaces. Early versions of such methods employed LSTMs (Lowe et al., 2015) but more recently, pre-trained models have been used (Karpukhin et al., 2020; Lu et al., 2020; Reimers and Gurevych, 2019; Liu et al., 2021). Models such as DPR (Karpukhin et al., 2020), S-BERT (Reimers and Gurevych, 2019) encode contexts and responses using dual encoders based on the BERT (Devlin et al., 2018) pre-trained model, and learn

a scoring function using negative samples. Models such as Poly-Encoder (Humeau et al., 2019), MEBERT (Luan et al., 2021b), ColBERT(Khattab and Zaharia, 2020) use multiple representations for dialog contexts instead of using a single representation.

Joint Encoding: In contrast to methods that independently encode context and response pairs, methods such as Sequential Matching Networks (Wu et al., 2017b), cross encoders using BERT (Nogueira and Cho, 2019; Chen et al., 2021b) jointly encode context and dialog responses. However, such models are slow during inference because all candidate responses need to be jointly encoded with the dialog context for scoring at runtime. This is in contrast to dual-encoder architectures where response embeddings can be computed offline and cached for efficient retrieval. Models such as ConvRT (Vakili Tahami et al., 2020), TwinBERT (Lu et al., 2020) use distillation to train a dual encoder from a cross encoder models to help a train better dual-encoder model.

3.2 Probabilistic Embeddings

Probabilistic embeddings have been applied in tasks for building better word representations (Qian et al., 2021; Athiwaratkun et al., 2018), entity comparison (Contractor et al., 2016), facial recognition (Chen et al., 2021a), pose estimation (Sun et al., 2020), generating multimodal embeddings (Athiwaratkun and Wilson, 2017; Chun et al., 2021), etc. The motivation in some of these tasks is similar to ours – for instance, Qian et al. (2021) use Gaussian embeddings to represent words to better capture meaning and ambiguity. However, to the best of our knowledge, the problem of applying probabilistic embeddings in dialog modeling tasks hasn't been explored. In this work, we represent dialog contexts as Mixture of Gaussians present approximate closed form expressions for efficiently computing KL-divergence based distance measures, thereby making it suitable for use in real-world settings.

4 Experiments

We answer the following questions through our experiments: (1) How does our model compare with recent dual-encoder based retrieval systems for the task of response retrieval? (2) Are the responses retrieved by our model more relevant and diverse? (3) Do human users of our system notice a difference in quality of response as compared to the recent, ColBERT system?

Due to space limitations, we answer the following questions in the Appendix: 1) Is the improvement in retrieval performance a consequence of the extra learnable parameters in Mix-and-Match? 2) How does the performance of Mix-and-Match depend on the number of Gaussian components in response and context GMM?

The model and training details are provided in the Appendix.

4.1 Datasets

We conduct our experiments on two publicly available datasets – Ubuntu Dialogue Corpus (Lowe et al., 2015)(v2.0)[4] and the Twitter Customer Support Dataset[5], and an internal technical support dataset. The Ubuntu Dialog Corpus v2.0 contains $500K$ context-response pairs in the training set and $20K$ context-response pairs in the validation set and test set respectively. The conversations deal with technical support for issues faced by Ubuntu users. The Twitter Customer Support Dataset contains $\sim$ 1 million context-response pairs in the training data and $\sim 120K$ context-response pairs in validation and test sets. The conversations deal with customer support provided by several companies on Twitter.

We also conduct our experiments on an internal real-world technical support dataset with $\sim 127K$ conversations. We will refer to this dataset as 'Tech Support dataset' in the rest of the paper. The Tech Support dataset contains conversations pertaining to an employee seeking assistance from an agent (technical support) — to resolve problems such as password reset, software installation/licensing, and wireless access. In contrast to Ubuntu dataset, which used user forums to construct the data, this dataset has clearly two distinct users — employee and agent. In all our experiments, we model the *agent* response turns only.

For each conversation in the Tech Support dataset, we sample context and response pairs. Note that multiple context-response pairs can be generated from a single conversation. We create validation pairs by selecting 5000 conversations randomly and sampling their context response pairs. Similarly, we create test pairs from a different subset of 5000 conversations. The remaining conver-

[4]https://github.com/rkadlec/ubuntu-ranking-dataset-creator

[5]https://www.kaggle.com/thoughtvector/customer-support-on-twitter

sations are used to create training context-response pairs.

4.2 Baselines

We compare our proposed model against two scalable baselines - SBERT (Reimers and Gurevych, 2019) and ColBERT (Khattab and Zaharia, 2020) – a recent state-of-the-art retrieval model. Similar to Mix-and-Match, both the baselines use independent encoders (dual-encoders to encode the contexts and responses. Hence, these baselines can be used for large-scale retrieval at an acceptable cost.

4.2.1 SBERT

SBERT (Reimers and Gurevych, 2019) uses two BERT encoders for embedding the inputs (context and response). We pass the contextualized embeddings at the last layer of BERT through the ReLU non-linearity followed by a linear layer to project it to a d-dimensional space. The projected embeddings are average-pooled to generate fixed size embeddings for context and response. Since context and response are from two different domains, we found that it is crucial that the context and response encoders do not share the parameters. We use inner-product between the context and response embeddings as the similarity measure and train the two encoders via contrastive loss.

4.2.2 ColBERT

Just like SBERT, ColBERT (Khattab and Zaharia, 2020) uses two BERT encoders to encode the inputs and pass the output through a linear layer to generate d-dimensional embeddings. However, instead of pooling the output through the linear layer, a late interaction is computed between all the contextualized token embeddings of the context and response. Unlike the original implementation of ColBERT, we do not enforce the context and response encoders to share parameters. This is essential for achieving reasonable performance for dialogs. The model is trained via contrastive loss. Please refer to the appendix for additional details about training and hyperparameter settings.

4.3 Response Retrieval

In this setting, each context is paired with 5000 randomly selected responses along with the ground truth response for the given context. The list of 5000 responses are randomly selected from the test data for each instance. Hence, the response universe associated with each dialog-context may be different. The task then is to retrieve the ground truth response given the context. For efficient computation, all the responses in the test data are encoded once and stored. Note that this is only possible for dual-encoder architectures (such as Mix-and-Match, SBERT, ColBERT); the major performance bottleneck in cross-encoder approaches arises from this step where the response encodings are dependent on the context and hence need to be encoded each time for every new dialog context.

For Mix-and-Match, the response encoder outputs the means and variances of the GMM induced by the response in the embedding space. We use a batch-size of 50 to encode the responses and cache the generated parameters (mean and variance) of the response-GMMs.

Similarly, the context is encoded by the context encoder to output the means and variances of the components of context-GMM. We compute the KL divergence between the context distribution and distribution of each response in the associated list of 5000 responses using the expressions derived in (6) and (7). The values are sorted in ascending order and the top-k responses are selected for evaluation.

A similar setting is used for SBERT and ColBERT with the exception that the embeddings are stored instead of means and variances. Moreover, we sort the responses based on SBERT and ColBERT similarity in descending order.

4.3.1 Results

We use MRR and Recall@k for evaluating the various models. For evaluating MRR, we sort the associated set of 5000 responses with each context, based on KL divergence in ascending order. For Recall@k, we pick the top-k responses with the least KL divergence. The percentage of contexts for which the ground truth response is present in the top-k responses is referred to as Recall@k. The results are shown in Table 1. For Mix-and-Match, we discovered that the optimal recall occurs when the number of Gaussian components in the GMM is small. The variation of performance with the number of Gaussian components is given in the appendix.

As can be observed, SBERT that uses a single embedding to represent the entire context as well as response, achieves the lowest recall. By using all the token embeddings to represent the context and response, ColBERT achieves better performance than SBERT. Finally, by using Gaussian mixture probability distributions to represent con-

Dataset	Model	Recall@1	Recall@2	Recall@5	Recall@10	MRR
Ubuntu (v2)	SBERT	6.24	8.44	13.26	18.26	0.099
	ColBERT	7.48	10.93	16.37	21.33	0.123
	Mix-and-Match (K=L=2)	**9.47**	**13.55**	**19.89**	**25.93**	**0.151**
Twitter	SBERT	6.87	9.82	19.08	29.64	0.135
	ColBERT	8.43	12.62	20.36	34.82	0.137
	Mix-and-Match (K=L=2)	**11.88**	**18.78**	**32.12**	**44.58**	**0.222**
Tech Support	SBERT	5.88	7.71	12.69	22.67	0.119
	ColBERT	**6.32**	8.82	14.97	23.91	0.125
	Mix-and-Match (K=L=2)	**6.73**	**9.67**	**15.68**	**26.47**	**0.133**

Table 1: Comparison of Mix-and-Match against baselines on retrieval tasks. Given a context, the task involves retrieving from a set of 5000 responses that also contains the ground truth response. The number of Gaussian components in the GMM are provided in paranthesis.

Model	Recall@5	MRR
SBERT+cross-encoder	20.91	0.155
ColBERT+cross-encoder	22.40	0.170
Mix-and-Match+cross-encoder	24.40	0.192

Table 2: Comparison of Mix-and-Match against baselines when coupled with cross-encoder on Ubuntu dataset.

text and response, Mix-and-Match achieves substantial improvement in Recall@k and MRR on all the datasets as compared to SBERT and ColBERT. Thus, richer the representation of context and response, better is the recall. In the appendix, we also include performance comparisons when the embedding size for SBERT and ColBERT is doubled. Note that the relative improvement is less in Tech Support, as there is less diversity among the responses in the training data of Tech Support. The agents are trained to handle calls in specific way that reduces the diversity.

Re-ranking with cross-encoder Instead of using the models above (SBERT, ColBERT, Mix-and-Match) for selecting a response, one may use these models to filter a subset of responses. The filtered responses can then be re-ranked using a more powerful, albeit slow models such as cross-encoders. In Table 2, we use a BERT-based cross-encoder (Nogueira and Cho, 2019; Chen et al., 2021b) for re-ranking the top-100 responses retrieved by each of the models on Ubuntu dataset. As can be observed, the scores of all the models improve significantly after re-ranking with cross-encoder. Moreover, the scores achieved by Mix-and-Match are significantly higher than the other baselines.

4.4 Response Recommendation

The response retrieval setting described in the previous section is unrealistic since it assumes that the ground truth response is also present in a set of 5000 responses. In reality, when a response retrieval model such as (Fadnis et al., 2020) is deployed for response recommendation, it must retrieve from a large set of all the responses present in the training data (often running into hundreds of thousands of responses).

To deal with the large set of responses present in the training data, we encode them offline using the response encoder of Mix-and-Match. As in the previous section, we use a batch-size of 50 for encoding the responses. After the means and variances of all the Gaussian components of response GMMs have been generated, we save them to a file along with the corresponding responses. To ensure faster retrieval, we use Faiss (Johnson et al., 2019) for indexing the means of the Gaussian components of response GMMs. Faiss is a library for computing fast vector-similarities and has been used for vector-based searching in huge sets. We use the IVFPQ index of faiss (Inverted File with Product Quantization) that discretizes the embedding space into a finite number of cells. This allows for faster search computations.

We flatten the tensor of means of Gaussian components of all response GMMs to a matrix of mean vectors. The matrix of mean vectors is added to the IVFPQ index. A pointer is maintained from the mean of each Gaussian component to the corresponding response as well as the means and variances of its Gaussian components.

When a new context arrives, we compute the means and variances of its Gaussian components. For each Gaussian component, we retrieve the top-10 responses by using the mean of the Gaussian component as the search query. After retrieving the top-10 responses for each Gaussian component, we load the corresponding means and variance. Finally, we compute the KL divergence between the context GMM and the GMMs of all the retrieved responses. The values are sorted in ascending order and the top-k responses are selected for evaluation.

Dataset	Model	BLEU-2	BLEU-4	Diversity (BERTDist.)
Ubuntu (v2)	SBERT	5.86	0.49	2.33
	ColBERT	6.66	0.58	3.19
	Mix-and-Match	**7.16**	**0.64**	**3.60**
Twitter	SBERT	19.84	10.3	1.76
	ColBERT	20.67	11.09	2.17
	Mix-and-Match	**22.83**	**12.62**	**2.60**
Tech Support	SBERT	12.09	5.82	1.49
	ColBERT	16.57	8.58	2.55
	Mix-and-Match	**18.82**	**10.57**	**3.02**

Table 3: Comparison of Mix-and-Match against baselines for the response recommendation task. Given a context, the task involves retrieving from the set of all responses in the training data. The computation of diversity is discussed in detail in Section 4.4

	ColBERT Win	Mix and Match Win	Tie
Response Relevance @1	17%	**40%**	43%
Diversity	42%	**58%**	NA

Table 4: The top-response returned by the Mix-and-match model is found to be relevant more often (40% vs 17%) than ColBERT. In addition, the set of responses returned by Mix-and-Match are also more diverse (58% vs 42% for ColBERT).

	ColBERT	Mix and Match
Diversified-Relevance (DR)	0.25	**0.35**

Table 5: The Diversified-Relevance scores for ColBERT and Mix-and-Match in our human study.

Language Generation Quality: Since the ground truth response may not be present verbatim in the set, metrics such as recall and MRR cannot be computed in this setting. We therefore use the BLEU metric (Papineni et al., 2002) for evaluating the quality of the responses. As can be observed from the table, the BLEU scores are quite low for Ubuntu dataset, suggesting that most retrieved responses have very little overlap with the ground truth response. As in the previous section, SBERT is outperformed by ColBERT in terms of BLEU-2 and BLEU-4. Finally, Mix-and-Match outperforms both the models on all three datasets. This suggests that the responses retrieved by Mix-and-Match are relevant to the dialog context.

Diversity of Responses: The primary strength of the Mix-and-Match system is its capability to associate multiple diverse responses with the same context. To capture the diversity among the top-k responses retrieved for a given context, we measure the distance between every pair of responses and average it across all pairs. Thus, if $\mathcal{R}$ is the set of retrieved responses for a given context, the BERT distance among the responses in $\mathcal{R}$ is given by

$$\text{BERTDistance}(\mathcal{R}) = \frac{1}{|\mathcal{R}|^2} \sum_{\mathbf{r} \in \mathcal{R}} \sum_{\bar{\mathbf{r}} \in \mathcal{R}} ||e(\mathbf{r}) - e(\bar{\mathbf{r}})||^2, \quad (9)$$

where $e(\bar{\mathbf{r}})$ is the pooled BERT embedding of $\mathbf{r}$.

The results are shown in Table 3. As can be observed from the table, SBERT has the least diversity among the retrieved responses. This is expected since all the retrieved responses must be close to the context embedding and hence, close to each other. ColBERT fares better in terms of diversity since it uses multiple embeddings to represent contexts and responses. Finally, Mix-and-Match that uses GMMs to represent contexts and responses achieves the best diversity. This suggests that having multiple or probabilistic embeddings helps in improving the diversity among the retrieved responses.

Scalability: Next, we evaluate the time taken by the Mix-and-Match model to retrieve from the FAISS index as comapred to baselines. The similarity/KL-divergence computations as well as vector similarity searches for the FAISS index, are performed on a single A100 GPU. Unsurprisingly, SBERT achieves the lowest latency of 8.9 ms for retrieval per dialog context. ColBERT achieves a latency of 89.7 ms. The latency of Mix-and-Match ranges from 36.7 ms (for 1 Gaussian component) to 68.8 ms (for 32 Gaussian components) depending upon the number of Gaussian components in the mixture. Note that, even in the worst case, the latency is less than 0.01s, thus making the model suitable for practical use in the real world.

Qualitative Study: Table 6 shows a sample with a multi-turn dialog context where the user is complaining about bad cellphone coverage. As before, the responses retrieved by both ColBERT and Mix-and-Match are presented. As can be seen, Mix-and-Match returns a relevant response at the top ranked position (highlighted in **green**) and related responses at other positions. In contrast, ColBERT retrieved generic or unrelated responses.

Human Evaluation: We also conducted a human study comparing the output responses of ColBERT and Mix-and-match. We used samples from the Twitter data set for this study as it does not require domain expertise to assess the relevance of responses. Three users were asked to review 30 twitter dialogs contexts along with the top-4 responses returned by each system,[6] in a response recommendation setting. Users were presented the

[6]a total of 360 independent context-response assessments.

Dialog Context	
User: the worst mobile serive in 2015 2017 cellphone badservice miami florida **Agent:** hey send us a dm and we'll ensure a great experience channeyt **User:** *tmobilehelp poor service low signal slow service it s miami*	
Responses Retrieved	
ColBERT	**Mix-and-Match**
(i) Our apologies , we are currently experiencing a system challenge which we are working to resolve . kindly bear with us.	**(i) how long has this been happening ? what type of phone do you have ? please send us a dm so we can fix it . thank you**
(ii) our sincere apologies for any inconveniences caused, we are having a technical issue, resolution is underway	(ii) that ' s not good at all ! please dm us with your zip code and nearest streets intersection to check the coverage
(iii) it is not our intention to make you upset. please feel free to reach out to us if you have already called back and still need further assistance.	(iii) does this happen in specific locations ? when did you begin to experience these issues with your connection ? are you having issues making calls and sending text as well ?
Ground Truth Response: let ' s flip thing around ! meet in the dms https://t.co/sbivwmm6x2	

Table 6: Sample of a multi-turn dialog context - Mix-and-Match returns a relevant response at the top ranked position and related responses at other positions. In contrast, ColBERT retrieved generic or unrelated responses.

outputs from each system in random order and they were blind to the system returning the responses. We asked our users the following:

1. Given the dialog context and the response sets from two different systems, label each response with a "yes" or "no' depending on whether the response is a relevant response recommendation for the dialog context. Thus, each response returned by both systems was individually labeled by three human users.

2. Given the dialog context and the response sets from two different systems, which of the response set is more diverse? Thus, each context-recommendation set was assessed by three human users.

We count the number of votes received by the top-ranked response for each system and report percentage wins for each system. In addition, we also report a head-to-head comparison in which the two models were assessed for diversity (no ties). Finally, to assess whether diversity is accompanied by relevance in the response set, we define a metric called *Diversified-Relevance (DR)* which weighs the diversity wins by the number of relevant responses returned by each system. Specifically, DR^{model}, the Diversified-Relevance for a $model \in$ {ColBERT, Mix-and-Match} is given by:

$$DR^{model} = \frac{\sum_i^M \sum_j^4 \mathbb{1}\{win_i^{model}\} * \mathbb{1}\{relevance_{ij}^{model}\}}{4M}, \quad (10)$$

where M is the number of dialogs used in the study, 4, is the number of response recommendations per dialog, $\mathbb{1}\{win_i^{model}\}$ is an indicator function that takes the value 1 if $model$ was voted as being more diverse its responses to i^{th} dialog context, and $\mathbb{1}\{relevance_{ij}^{model}\}$, is an indicator function that takes the value 1 if the j^{th} response recommendation by $model$ was voted as being relevant[7].

As can be seen in Table 4, the top-ranked response returned by Mix-and-Match received significantly higher number of votes (40%) in favour as compared to ColBERT. In 43% of the cases there was no-clear winner. Finally, in 58% of the dialogs, Mix-and-Match was found to present a more diverse set of response recommendations.

In order to assess, if the diversity is accompanied by relevance, we also report the DR scores in Table 5. As can be seen the DR scores for Mix-and-Match is significantly higher than ColBERT (0.35 vs 0.25). Overall, the results from our human-study indicate that Mix-and-Match returns more diverse and relevant responses.

5 Conclusion

By modelling contexts and responses as multi-modal distributions, we allow the network to be more expressive without forcing the representations of unrelated responses to move closer, as would have been the case with traditional dual-encoder learning objectives. We derived and presented a closed form expressions for efficiently computing the KL-divergence based distance measures and showed its suitability for real-world settings. We demonstrated the effectiveness of our retrieval systems on three different datasets - Ubuntu, Twitter and an internal, real-world Tech support dataset. Additional experiments for response relevance, including a human study were performed on the publicly available datasets. We found that not only is our model able to retrieve more relevant responses as compared to recent retrieval systems, it also presented more diverse results.

[7] As can be seen DR returns a score between 0 and 1.

6 Limitations

Mix-and-Match relies heavily on the diversity of responses for a given input for achieving good performance. As a result, the model doesn't achieve significant performance boost when the diversity isn't significant. In our experiments, we observed this trend for our internal Tech Support dataset which had standard responses for most queries.

Moreover, it isn't straightforward to store the context and response GMMs in the nearest neighbor index. In our experiments, we used a workaround where we store the means of all the Gaussian components in the nearest neighbor index. To retrieve, we used the L2 distance between the means of the Gaussian components of response and context GMMs. The retrieved results were then reranked using the KL-divergence approximation discussed in the paper. By ignoring the variance term, we are forced to assume that the Gaussian components in the GMMs are spherical.

References

Johan Aronsson, Philip Lu, Daniel Strüber, and Thorsten Berger. 2021. A maturity assessment framework for conversational ai development platforms. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, pages 1736–1745.

Ben Athiwaratkun and Andrew Wilson. 2017. Multimodal word distributions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1645–1656, Vancouver, Canada. Association for Computational Linguistics.

Ben Athiwaratkun, Andrew Wilson, and Anima Anandkumar. 2018. Probabilistic FastText for multi-sense word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–11, Melbourne, Australia. Association for Computational Linguistics.

Alexander Bartl and Gerasimos Spanakis. 2017. A retrieval-based dialogue system utilizing utterance and context embeddings. *CoRR*, abs/1710.05780.

Jane Bromley, James W Bentz, Léon Bottou, Isabelle Guyon, Yann LeCun, Cliff Moore, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a "siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04):669–688.

Kai Chen, Qi Lv, Taihe Yi, and Zhengming Yi. 2021a. Reliable probabilistic face embeddings in the wild. *CoRR*, abs/2102.04075.

Xiaoyang Chen, Kai Hui, Ben He, Xianpei Han, Le Sun, and Zheng Ye. 2021b. Co-bert: A context-aware bert retrieval model incorporating local and query-specific context. *arXiv preprint arXiv:2104.08523*.

Sanghyuk Chun, Seong Joon Oh, Rafael Sampaio de Rezende, Yannis Kalantidis, and Diane Larlus. 2021. Probabilistic embeddings for cross-modal retrieval. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8411–8420.

Danish Contractor, Parag Singla, and Mausam. 2016. Entity-balanced Gaussian pLSA for automated comparison. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 69–79, San Diego, California. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. Cite arxiv:1810.04805Comment: 13 pages.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Pankaj Dhoolia, Vineet Kumar, Danish Contractor, and Sachindra Joshi. 2021. Bootstrapping dialog models from human to human conversation logs. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 16024–16025. AAAI Press.

Kshitij P. Fadnis, Nathaniel Mills, Jatin Ganhotra, Haggai Roitman, Gaurav Pandey, Doron Cohen, Yosi Mass, Shai Erera, R. Chulaka Gunasekara, Danish Contractor, Siva Sankalp Patel, Q. Vera Liao, Sachindra Joshi, Luis A. Lastras, and David Konopnicki. 2020. Agent assist through conversation analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 151–157. Association for Computational Linguistics.

Jia-Chen Gu, Tianda Li, Quan Liu, Zhen-Hua Ling, Zhiming Su, Si Wei, and Xiaodan Zhu. 2020. Speaker-aware bert for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2041–2044.

Janghoon Han, Taesuk Hong, Byoungjae Kim, Youngjoong Ko, and Jungyun Seo. 2021. Fine-grained post-training for improving retrieval-based

dialogue systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1549–1558.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. *arXiv: Computation and Language*.

Zongcheng Ji, Zhengdong Lu, and Hang Li. 2014. An information retrieval approach to short text conversation. *CoRR*, abs/1408.6988.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Peiyang Liu, Sen Wang, Xi Wang, Wei Ye, and Shikun Zhang. 2021. QuadrupletBERT: An efficient model for embedding-based large-scale retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3734–3739, Online. Association for Computational Linguistics.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *arXiv preprint arXiv:1506.08909*.

Wenhao Lu, Jian Jiao, and Ruofei Zhang. 2020. Twinbert: Distilling knowledge to twin-structured bert models for efficient retrieval. *arXiv preprint arXiv:2002.06275*.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021a. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021b. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *CoRR*, abs/1901.04085.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Chen Qian, Fuli Feng, Lijie Wen, and Tat-Seng Chua. 2021. Conceptualized and contextualized gaussian embedding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13683–13691.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems*, pages 1857–1865.

Yixuan Su, Deng Cai, Qingyu Zhou, Zibo Lin, Simon Baker, Yunbo Cao, Shuming Shi, Nigel Collier, and Yan Wang. 2021. Dialogue response selection with hierarchical curriculum learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1740–1751.

Jennifer J. Sun, Jiaping Zhao, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, and Ting Liu. 2020. View-invariant probabilistic embedding for human pose. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V*, volume 12350 of *Lecture Notes in Computer Science*, pages 53–70. Springer.

Amir Vakili Tahami, Kamyar Ghajar, and Azadeh Shakery. 2020. *Distilling Knowledge for Fast Retrieval-Based Chat-Bots*, page 2081–2084. Association for Computing Machinery, New York, NY, USA.

Taesun Whang, Dongyub Lee, Dongsuk Oh, Chanhee Lee, Kijong Han, Dong-hun Lee, and Saebyeok Lee. 2021. Do response selection models really know what's next? utterance manipulation strategies for multi-turn response selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14041–14049.

Yu Wu, Wei Wu, Chen Xing, Can Xu, Zhoujun Li, and Ming Zhou. 2017a. A sequential matching framework for multi-turn response selection in retrieval-based chatbots. *CoRR*, abs/1710.11344.

Yu Wu, Wei Wu, Ming Zhou, and Zhoujun Li. 2017b. Sequential match network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *ACL*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Ruijian Xu, Chongyang Tao, Daxin Jiang, Xueliang Zhao, Dongyan Zhao, and Rui Yan. 2021. Learning an effective context-response matching model with self-supervised tasks for retrieval-based dialogues. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14158–14166.

Rui Yan, Yiping Song, and Hua Wu. 2016. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *SIGIR*.

Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. *Few-Shot Conversational Dense Retrieval*, page 829–838. Association for Computing Machinery, New York, NY, USA.

Supplementary material for Mix-and-Match: Scalable Dialog Response Retrieval using Gaussian Mixture Embeddings

1 Proof of Theorem 1

Proof. The proof follows a similar line of reasoning as the proof provided in (Hershey and Olsen, 2007) The KL divergence between p_r and p_c can be written as

$$KL(p_r||p_c) = \int p_r(z) \log p_r(z) \mathrm{d}z \quad (1)$$

$$- \int p_r(z) \log p_c(z) \mathrm{d}z$$

$$= -\mathcal{H}(p_r) + \mathcal{H}(p_r, p_c) \quad (2)$$

The first term is the negative of entropy while the second term is the cross entropy. We approximate the cross entropy by expanding the GMM in terms of its Gaussian components, and applying Jensen's inequality:

$$\mathcal{H}(p_r, p_c)$$

$$= -\frac{1}{L} \sum_{\ell=1}^{L} \int p_r(z;\ell) \log \left[\sum_{k=1}^{K} q_\ell(k) \frac{p_c(z;k)}{q_\ell(k) K} \right] \mathrm{d}z$$

$$\leq -\frac{1}{L} \sum_{\ell=1}^{L} \sum_{k=1}^{K} q_\ell(k) \int p_r(z;\ell) \log p_c(z;k) \mathrm{d}z$$

$$\frac{1}{L} \sum_{\ell=1}^{L} \sum_{k=1}^{K} q_\ell(k) \log q_\ell(k) + \log K$$

$$= \frac{1}{L} \sum_{\ell=1}^{L} \sum_{k=1}^{K} q_\ell(k) \mathcal{H}(p_r(.;\ell), p_c(.;k)) \quad (3)$$

$$- \mathcal{H}(q_\ell) + \log K \quad (4)$$

Here, the first equality follows by multiplying and dividing the terms within the log by the variational distribution $q_\ell(k)$. The last inequality follows by applying Jensen's inequality. The above upper bound holds for all choice of q. The bound can be tightened by minimizing it with respect to $q_\ell(k)$. We assume q_ℓ to be a one-hot vector which can only be non-zero for one context component k. Every one-hot q_ℓ has an entropy of 0 and hence, the second term in the equation is always 0. For a one-hot q_ℓ, the above equation is minimized when q_ℓ assigns all its weights to the component of context GMM with lowest cross-entropy. Using the optimal one-hot q, the above equation can be written as

$$\mathcal{H}(p_r, p_c) \leq \quad (5)$$

$$\frac{1}{L} \sum_{\ell=1}^{L} \min_{k \in \{1,\ldots,K\}} \mathcal{H}(p_r(.;\ell), p_c(.;k)) + \log K \quad (6)$$

The entropy of p_r can be derived as a special case of the above equation by replacing p_c in the above equation by p_r. Thus, the entropy of a GMM can be upper-bounded by

$$\mathcal{H}(p_r) \leq \frac{1}{L} \sum_{\ell=1}^{L} \mathcal{H}(p_r(.;\ell)) + \log L \quad (7)$$

Finally, the KL divergence can be approximated by replacing (6) and (7) in (2). Note that the resultant quantity is neither an upper nor a lower bound, but still a useful approximation.

$$\mathrm{KL}(p_r||p_c) \quad (8)$$

$$\approx \frac{1}{L} \sum_{\ell=1}^{L} \min_{k \in \{1,\ldots,K\}} [\mathcal{H}(p_r(.;\ell), p_c(.;k))] \quad (9)$$

$$+ \frac{1}{L} \sum_{\ell=1}^{L} -\mathcal{H}(p_r(.;\ell)) + \log(K/L) \quad (10)$$

$$= \frac{1}{L} \sum_{\ell=1}^{L} \min_{k \in \{1,\ldots,K\}} \mathrm{KL}(p_r(.;\ell)||p_c(.;k)) \quad (11)$$

$$+ \log(K/L) \quad (12)$$

□

2 Model and training details

We ran all our experiments on a single Nvidia A100 GPU. We use the pretrained 'bert-base' model pro-

vided by Hugging Face[1]. The dimension of the embedding space is fixed to be 128 for all the models. The number of Gaussian components in the context and response distributions is selected by cross-validation from the set $\{1, 2, 4, 8, 16, 32\}$. We use the 'AdamW' optimizer provided by Hugging Face (Adam optimizer with a fixed weight decay) with a learning rate of $1.5e-5$ for all our experiments. A fixed batch size of 16 context-response pairs is used. To prevent overfitting, we use early-stopping with the loss function defined in Section 2.6 on validation set as the stopping criteria. The time taken by the various algorithms to reach convergence on Ubuntu dataset is as follows: 37.5 hours for SBERT, 38.5 hours for ColBERT and 38.8 hours for Mix-and-Match with $K = L = 2$. The total number of parameters in each model (including the baseline) is approximately 219 million.

3 Ablation studies

In our ablation studies, we want to answer the following questions: 1) Is the improvement in retrieval performance a consequence of the extra learnable parameters in Mix-and-Match? 2) How does the performance of Mix-and-Match depend on the number of Gaussian components in response and context GMM?

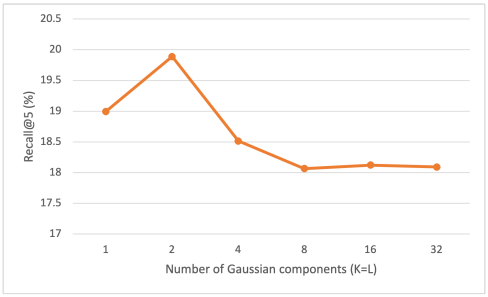
3.1 Effect of the extra parameters in Mix-and-Match

All the three models used in our experiments (SBERT, ColBERT and Mix-and-Match) use the same BERT architecture for encoding. However, for SBERT and ColBERT, the BERT encodings are passed through a single linear layer for generating the final embeddings. Instead, for Mix-and-Match, two parallel linear layers are used to generate the means and log-variances of the Gaussian mixtures. These layers are shared by all the Gaussian components in the mixture.

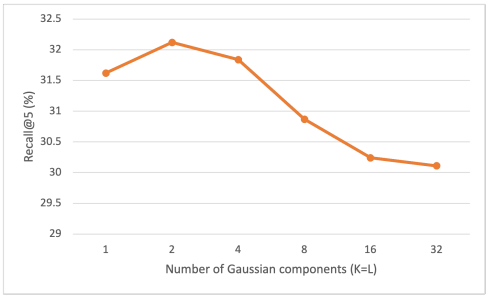
To account for this extra linear layer, we double the embedding size for SBERT and ColBERT. The resultant retrieval scores are present in Table **??**. As can be observed from the table, the retrieval performance of SBERT and ColBERT improves by doubling the embedding layer. However, Mix-and-Match with half the embedding size (size of the mean vector) still outperforms these baselines on Twitter and Tech support dataset. On Ubuntu dataset, the performance of the various models is comparable.

3.2 Variation of retrieval accuracy with the number of Gaussian components

Next, we plot the retrieval accuracy of Mix-and-Match with the number of Gaussian components in the GMM. We vary the number of Gaussian components in the context and response GMM from 1 to 32 and compute the recall@5 on Ubuntu and Twitter dataset. As can be observed from Figure **??**, the recall is high when the number of Gaussian components is small but starts decreasing as we increase the number of components. Overall, the best performance occurs when $K = L = 2$ or $K = L = 4$.



(a) Ubuntu (v2)



(b) Twitter

Figure 1: Variation of recall@5 with the number of Gaussian components in the context and response GMM.

3.3 Qualitative Study

Table 2 shows a sample with a single-turn dialog context where the user is complaining about flight boarding positions. The responses retrieved by both ColBERT and Mix-and-Match are presented. As can be seen, Mix-and-Match returns a relevant response at the top ranked position (highlighted in **green**) and another related response at the second position. In contrast, ColBERT retrieved generic or unrelated responses.

[1]https://huggingface.co/bert-base-uncased

Table 1: Comparison of Mix-and-Match against baselines with twice the embedding size on retrieval tasks. Given a context, the task involves retrieving from a set of 5000 responses that also contains the ground truth response.

Dataset	Model	Recall@2	Recall@5	Recall@10	MRR
Ubuntu v2	SBERT (256)	9.73	15.62	21.10	0.114
	ColBERT (256)	**13.60**	**19.60**	24.75	**0.150**
	Mix-and-Match (128)	**13.55**	**19.89**	**25.93**	**0.151**
Twitter	SBERT (256)	11.08	20.8	31.8	0.144
	ColBERT (256)	12.98	21.94	35.78	0.149
	Mix-and-Match (128)	**18.78**	**32.12**	**44.58**	**0.222**
Tech Support	SBERT (256)	8.11	13.43	23.86	0.123
	ColBERT (256)	9.53	15.42	25.17	0.131
	Mix-and-Match (128)	**9.67**	**15.68**	**26.47**	**0.133**

Table 2: Sample of a single-turn dialog context - Mix-and-Match returns a relevant response at the top ranked position and another related response at the second position. In contrast, ColBERT retrieved generic or unrelated responses.

Dialog Context	
User: *@southwestair i'm bummed i paid 15 for early bird and still only got a b boarding position don t think i ll do that anymore notworthit*	
Responses Retrieved	
ColBERT	**Mix-and-Match**
(i)if you check your itinerary email, it ' ll tell you what type of fare you purchased.wanna get away fares are nonrefundable, but anytime and business select fares can be refunded. thanks for reaching out	**(i) please know that earlybird offers the convenience of automatic check - in, but does not guarantee a specific boarding position . still, we apologize if you were disappointed in the position you received &hope to make it up to you while onboard**
(ii) sorry if we've missed a chance to be of better service to you. since we're separate entities, we're re unable to alter or change the incorrect alamo car reservation	(ii) by law we ' re required to let folks with specific boarding needs or disabilities on the aircraft first, though we apologize for your frustration this morning
(iii) oh no ! so sorry to hear that. please speak with our agents in the airport about reaccommodations	(iii) sorry for any confusion, our agents know the proper procedures and questions toask to determine the best boarding option
Ground Truth Response: apologies for any frustration, as the # of earlybird customers vary on each flt, you're guaranteed automatic check in, not a specific boarding position .	

References

John R Hershey and Peder A Olsen. 2007. Approximating the kullback leibler divergence between gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–317. IEEE.