Pretraining Methods for Dialog Context Representation Learning

Shikib Mehri*, **Evgeniia Razumovskaia***, **Tiancheng Zhao and Maxine Eskenazi**
Language Technologies Institute, Carnegie Mellon University
{amehri,erazumov,tianchez,max+}@cs.cmu.edu

Abstract

This paper examines various unsupervised pretraining objectives for learning dialog context representations. Two novel methods of pretraining dialog context encoders are proposed, and a total of four methods are examined. Each pretraining objective is fine-tuned and evaluated on a set of downstream dialog tasks using the MultiWoz dataset and strong performance improvement is observed. Further evaluation shows that our pretraining objectives result in not only better performance, but also better convergence, models that are less data hungry and have better domain generalizability.

1 Introduction

Learning meaningful representations of multi-turn dialog contexts is the cornerstone of dialog systems. In order to generate an appropriate response, a system must be able to aggregate information over multiple turns, such as estimating a belief state over user goals (Williams et al., 2013) and resolving anaphora co–references (Mitkov, 2014). In the past, significant effort has gone into developing better neural dialog architectures to improve context modeling given the same in-domain training data (Dhingra et al., 2017; Zhou et al., 2016). Recent advances in pretraining on massive amounts of text data have led to state-of-the-art results on a range of natural language processing (NLP) tasks (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2018) including natural language inference, question answering and text classification. These promising results suggest a new direction for improving context modeling by creating general purpose natural language representations that are useful for many different downstream tasks.

Yet pretraining methods are still in their infancy. We do not yet fully understand their properties. For example, many pretraining methods are variants of language modeling (Howard and Ruder, 2018; Radford et al., 2018; Devlin et al., 2018), e.g. predicting the previous word, next word or the masked word, given the sentence context. This approach treats natural language as a simple stream of word tokens. It relies on a complex model to discover high-level dependencies, through the use of massive corpora and expensive computation. Recently the BERT model (Devlin et al., 2018) achieved state-of-the-art performance on several NLP benchmarks. It introduces a sentence-pair level pretraining objective, i.e. predicting whether two sentences should come after one another. This is a step towards having pretraining objectives that explicitly consider and leverage discourse-level relationships. However, it is still unclear whether language modeling is the most effective method of pretrained language representation, especially for tasks that need to exploit multi-turn dependencies, e.g. dialog context modeling. Thornbury and Slade (2006) underline several discourse-level features which distinguish dialog from other types of text. Dialog must be coherent across utterance and a sequence of turns should achieve a communicative purpose. Further, dialog is interactive in nature, with feedback and back-channelling between speakers, and turn-taking. These unique features of dialog suggest that modelling dialog contexts requires pretraining methods specifically designed for dialog.

Building on this prior research, the goal of this paper is to study various methods of pretraining discourse-level language representations, i.e. modeling the relationship amongst multiple utterances. This paper takes a first step in the creation of a systematic analysis framework of pretraining methods for dialog systems. Concretely,

---

* Equal contribution.

we pretrain a hierarchical dialog encoder (Serban et al., 2016) with four different unsupervised pre-training objectives. Two of the objectives, next-utterance generation (Vinyals and Le, 2015) and retrieval (Lowe et al., 2016), have been explored in previous work. The other two pretraining objectives, masked-utterance retrieval and inconsistency identification, are novel. The pretrained dialog encoder is then evaluated on several downstream tasks that probe the quality of the learned context representation by following the typical pretrain & fine-tune procedure.

Pretraining and downstream evaluation use the MultiWoz dialog dataset (Budzianowski et al., 2018), which contains over 10,000 dialogs spanning 6 different domains. The downstream tasks include next-utterance generation (NUG), next-utterance retrieval (NUR), dialog act prediction (DAP), and belief state prediction (BSP). The pretraining objectives are assessed under four different hypotheses: (1) that pretraining will improve downstream tasks with fine-tuning on the entire available data, (2) that pretraining will result in better convergence, (3) that pretraining will perform strongly with limited data and (4) that pretraining facilitates domain generalizability. The results here show that pretraining achieves significant performance gains with respect to these hypotheses. Furthermore, the novel objectives achieve performance that is on-par with or better than the pre-existing methods. The contributions of this paper are: (1) a study of four different pretraining objectives for dialog context representation, including two novel objectives. (2) a comprehensive analysis of the effects of pretraining on dialog context representations, assessed on four different downstream tasks.

2 Related Work

This work is closely related to research in auxiliary multi-task learning and transfer learning with pretraining for NLP systems.

Training with Auxiliary Tasks

Incorporating a useful auxiliary loss function to complement the primary objective has been shown to improve the performance of deep neural network models, including, but not limited to, error detection (Rei and Yannakoudakis, 2017), cross-lingual speech tagging (Plank et al., 2016), domain independent sentiment classification (Yu and Jiang, 2016), latent variable inference for dialog generation (Zhao et al., 2017) and opinion extraction (Ding et al., 2017). Some auxiliary loss functions are designed to improve performance on a specific task. For instance, Yu and Jiang (2016) pretrained a model for sentiment classification with the auxiliary task of identifying whether a negative or positive word occurred in the sentence. In some cases, auxiliary loss is created to encourage a model's general representational power. Trinh et al. (2018) found that a model can capture far longer dependencies when pretrained with a suitable auxiliary task. This paper falls in line with the second goal by creating learning objectives that improve a representation to capture general-purpose information.

Transfer Learning with Pretraining

The second line of related research concerns the creation of transferable language representation via pretraining. The basic procedure is typically to first pretrain a powerful neural encoder on massive text data with unsupervised objectives. The second step is to fine-tune this pretrained model on a specific downstream task using a much smaller in-domain dataset (Howard and Ruder, 2018). Recently, several papers that use this approach have achieved significant results. ELMo (Peters et al., 2018) trained a two-way language model with Bidirectional Long Short-Term Memory Networks (biLSTM) (Huang et al., 2015) to predict both the next and previous word. OpenAI's GPT created a unidirectional language model using transformer networks (Radford et al., 2018) and BERT was trained with two simultaneous objectives: the masked language model and next sentence prediction (Devlin et al., 2018). Each of the models has demonstrated state-of-the-art results on the GLUE benchmark (Wang et al., 2018). The GPT model has also been adapted to improve the performance of end-to-end dialog models. In the 2nd ConvAI challenge (Dinan et al., 2019), the best models on both human and automated evaluations were generative transformers (Wolf et al., 2019), which were initialized with the weights of the GPT model and fine-tuned on in-domain dialog data. These models, which leveraged large-scale pretraining, outperformed the systems which only used in-domain data.

There has been little work on pretraining methods that learn to extract discourse level information from the input text. Next sentence predic-

tion loss in BERT ([Devlin et al., 2018](#)) is a step in this direction. While these pretraining methods excel at modelling sequential text, they do not explicitly consider the unique discourse-level features of dialog. We therefore take the first steps in the study of pretraining objectives that extract better discourse-level representations of dialog contexts.

3 Pretraining Objectives

This section discusses the unsupervised pretraining objectives, including two novel approaches aimed at capturing better representations of dialog context. When considering a specific pretraining method, both the pretraining objective and the model architecture must facilitate the learning of *strong* and *general* representations. We define a *strong* representation as one that captures the discourse-level information within the entire dialog history as well as utterance-level information in the utterances that constitute that history. By our definition, a representation is sufficiently *general* when it allows the model to perform better on a variety of downstream tasks. The next section describes the pretraining objectives within the context of the strength and generality of the learned representations.

For clarity of discussion, the following notation is used: an arbitrary T-turn dialog segment is represented by a list of utterances $c = [u_1, ...u_T]$, where u_i is an utterance. Further, we denote the set of all observed dialog responses in the data by $R = \{r_1, ...r_M\}$.

The pretraining objectives, discussed below, are next-utterance retrieval (NUR), next-utterance generation (NUG), masked-utterance retrieval (MUR), and inconsistency identification (InI).

3.1 Next-Utterance Retrieval

NUR has been extensively explored both as an independent task ([Lowe et al., 2015, 2016](#)) and as an auxiliary loss in a multi-tasking setup ([Wolf et al., 2019](#)). Given a dialog context, the aim of NUR is to select the correct next utterance from a set of k candidate responses. NUR can be thought of as being analogous to language modelling, except that the utterances, rather than the words, are the indivisible atomic units. Language modelling pretraining has produced strong representations of language ([Radford et al., 2018; Peters et al., 2018](#)), thereby motivating the choice of NUR as a pre-

training objective.

For this task we use a hierarchical encoder to produce a representation of the dialog context by first running each utterance independently through a Bidirectional Long-short Term Memory Network (biLSTM) and then using the resulting utterance representations to produce a representation of the entire dialog context. We use a single biLSTM to encode candidate responses. Given $[u_1, ...u_{T-1}]$, the task of NUR is to select the correct next utterance u_T from R. Note that for large dialog corpora, R is usually very large and it is more computationally feasible to sample a subset of R and as such we retrieve K negative samples for each training example, according to some distribution $p_n(r)$, e.g. uniform distribution ([Mikolov et al., 2013](#)). Concretely, we minimize the cross entropy loss of the next utterance by:

$$\hat{\mathbf{u}}_\mathbf{i} = f_\mathrm{u}(u_i) \quad i \in [1, T-1] \tag{1}$$

$$[\mathbf{h_1}, ...\mathbf{h_{T-1}}] = f_\mathrm{c}(\hat{\mathbf{u}}_\mathbf{1}, ...\hat{\mathbf{u}}_\mathbf{{T-1}}) \tag{2}$$

$$\mathbf{r_{gt}} = f_\mathrm{r}(u_T) \tag{3}$$

$$\mathbf{r_j} = f_\mathrm{r}(r_j) \quad r_j \sim p_n(r) \tag{4}$$

$$\alpha_{gt} = \mathbf{h_{T-1}}^T \mathbf{r_{gt}} \tag{5}$$

$$\alpha_j = \mathbf{h_{T-1}}^T \mathbf{r_j} \tag{6}$$

where f_u, f_c and f_r are three distinct biLSTM models that are to be trained. The final loss function is:

$$\mathcal{L} = -\log p(u_T|u_1, ...u_{T-1}) \tag{7}$$

$$= -\log \left(\frac{\exp(\alpha_{gt})}{\exp(\alpha_{gt}) + \sum_{j=1}^{K} \exp(\alpha_j)} \right)$$

3.2 Next-Utterance Generation

NUG is the task of generating the next utterance conditioned on the past dialog context. Sequence-to-sequence models ([Sutskever et al., 2014; Bahdanau et al., 2015](#)) have been used for pretraining ([Dai and Le, 2015; McCann et al., 2017](#)), and have been shown to learn representations that are useful for downstream tasks ([Adi et al., 2016; Belinkov et al., 2017](#)).

The hierarchical recurrent encoder-decoder architecture ([Serban et al., 2016](#)) was used during NUG pretraining. Although the decoder is used in pretraining, only the hierarchical context encoder is transferred to the downstream tasks. Similarly to NUR, the optimization goal of NUG is to maximize the log-likelihood of the next utterance given

the previous utterances. However, it differs in that it factors the conditional distribution to word-level in an auto-regressive manner. Specifically, let the word tokens in u_T be $[w_1, ...w_N]$. The dialog context is encoded as in Eq 8 with an utterance and a context biLSTM. Then the loss function to be minimized is shown in Eq 9:

$$\mathcal{L} = -\log p(u_T|u_1, ...u_{T-1}) \tag{8}$$

$$= -\sum_{k}^{N} \log p(w_k|w_{<k}, \mathbf{h_{T-1}}) \tag{9}$$

3.3 Masked-Utterance Retrieval

MUR is similar to NUR: the input contains a dialog context and a set of K candidate responses. The objective is to select the correct response. The difference between the two is twofold. First, one of the utterances in the dialog context has been replaced by a randomly chosen utterance. Secondly, rather than use the final context representation to select the response that should immediately follow, the goal here is to use the representation of the *replacement utterance* to retrieve the correct utterance. The replacement index t is randomly sampled from the dialog segment:

$$t \sim \text{Uniform}[1, T] \tag{10}$$

Then u_t is randomly replaced by a replacement utterance q that is sampled from the negative distribution $p_n(r)$ defined in NUR. Finally, the goal is to minimize the negative log-likelihood of the original u_t given the context hidden state at timestamp t, i.e. $-\log p(u_{gt}|u_1, ...q, ...u_T)$, where u_{gt} is the original utterance at index t.

$$\mathbf{\hat{u}_i} = f_{\text{u}}(u_i) \quad i \in [1, T] \tag{11}$$

$$[\mathbf{h_1}, ...\mathbf{h_T}] = f_{\text{c}}(\mathbf{\hat{u}_1}, ...\mathbf{\hat{u}_T}) \tag{12}$$

$$\mathbf{r_{gt}} = f_{\text{r}}(u_{gt}) \tag{13}$$

$$\mathbf{r_j} = f_{\text{r}}(r_j) \quad r_j \sim p_n(r) \tag{14}$$

$$\alpha_{gt} = \mathbf{h_t}^T \mathbf{r_{gt}} \tag{15}$$

$$\alpha_j = \mathbf{h_t}^T \mathbf{r_j} \tag{16}$$

The final loss function is:

$$\mathcal{L} = -\log p(u_t|u_1, ...q, ...u_T) \tag{17}$$

$$= -\log \left(\frac{\exp(\alpha_{gt})}{\exp(\alpha_{gt}) + \sum_{j=1}^{K} \exp(\alpha_j)} \right)$$

MUR is analogous to the MLM objective of Devlin et al. (2018), which forces model to keep a distributional contextual representation of each input token. By masking entire utterances, instead of input tokens, MUR learns to produce strong representations of each utterance.

3.4 Inconsistency Identification

InI is the task of finding inconsistent utterances within a dialog history. Given a dialog context with one utterance replaced randomly, just like MUR, InI finds the inconsistent utterance. The replacement procedure is the same as the one described for MUR, where a uniform random index t is selected in the dialog context and u_t is replaced by a negative sample q.

While MUR strives to create a model that finds the original utterance, given the replacement index t, InI aims to train a model that can identify the replacement position t. Specifically, this is done via:

$$\mathbf{\hat{u}_i} = f_{\text{u}}(u_i) \quad i \in [1, T] \tag{18}$$

$$[\mathbf{h_1}, ...\mathbf{h_T}] = f_{\text{c}}(\mathbf{\hat{u}_1}, ...\mathbf{\hat{u}_T}) \tag{19}$$

$$\alpha_i = \mathbf{h_T}^T \mathbf{h_i} \quad i \in [1, T] \tag{20}$$

Finally, the loss function is to minimize the cross entropy of the replaced index:

$$\mathcal{L} = -\log p(t|u_1, ...q, ...u_T) \tag{21}$$

$$= -\log \left(\frac{\exp(\alpha_t)}{\sum_{j=1}^{T} \exp(\alpha_i)} \right)$$

This pretraining objective aims to explicitly model the coherence of the dialog, which encourages both local representations of each individual utterance and a global representation of the dialog context. We believe that this will improve the generality of the pretrained representations.

4 Downstream Tasks

This section describes the downstream tasks chosen to test the strength and generality of the representations produced by the various pretraining objectives. The downstream evaluation is carried out on a lexicalized version of the MultiWoz dataset (Budzianowski et al., 2018). MultiWoz contains multi-domain conversations between a Wizard-of-Oz and a human. There are 8422 dialogs for training, 1000 for validation and 1000 for testing.

4.1 Belief State Prediction

Given a dialog context, the task is to predict a 1784-dimensional belief state vector. Belief state prediction (BSP) is a multi-class classification task, highly dependant on strong dialog context representations. The belief state vector represents the values of 27 entities, all of which can be inferred from the dialog context. To obtain the 1784-dimensional label, the entity values are encoded as a one-hot encoded vector and concatenated. The entities are shown in Appendix **??**. Performance is measured using the F-1 score for entities with non-empty values. This approach is analogous to the one used in the evaluation of Dialog State Tracking Challenge 2 (Henderson et al., 2014).

This task measures the ability of a system to maintain a complete and accurate state representation of the dialog context. With a 1784-dimensional output, the hidden representation for this task must be sufficiently general. Therefore, any pretrained representations that lack generality will struggle on belief state prediction.

4.2 Dialog Act Prediction

Dialog act prediction (DAP), much like belief state prediction, is a multi-label task aimed at producing a 32-dimensional dialog act vector for the system utterances. The set of dialog acts for a system utterance describes the actions that may be taken by the system. This might include: informing the user about an attraction, requesting information about a hotel query, or informing them about specific trains. There are often multiple actions taken in a single utterance, and thus this is a multi-label task. To evaluate performance on dialog act prediction, we use the F-1 score.

4.3 Next-Utterance Generation

NUG is the task of producing the next utterance conditioned on the dialog history. We evaluate the ability of our models to generate system utterances using BLEU-4 (Papineni et al., 2002). This task requires both a strong global context representation to initialize the decoder's hidden state and strong local utterance representations.

4.4 Next-Utterance Retrieval

Given a dialog context, NUR selects the correct next utterance from a set of k candidate responses. Though this task was not originally part of the MultiWoz dataset, we construct the necessary data

for this task by randomly sampling negative examples. This task is underlined by Lowe et al. (2016)'s suggestion that using NUR for evaluation is extremely indicative of performance and is one of the best forms of evaluation. Hits@1 (H@1) is used to evaluate our retrieval models. The latter is equivalent to accuracy.

Although some of these pretraining models had a response encoder, which would have been useful to transfer to this task, to ensure a fair comparison of all of the methods, we only transfer the weights of the context encoder.

5 Experiments and Results

This section presents the experiments and results aimed at capturing the capabilities and properties of the above pretraining objectives by evaluating on a variety of downstream tasks. All unsupervised pretraining objectives are trained on the full MultiWoz dataset (Budzianowski et al., 2018). Data usage for downstream fine-tuning differs, depending on the property being measured.

5.1 Experimental Setup

Each model was trained for 15 epochs, with the validation performance computed at each epoch. The model achieving the highest validation set performance was used for the results on the test data. The hyperparameters and experimental settings are shown in the Appendix **??**. The source code will be open-sourced when this paper is released.

In the experiments, the performance on each downstream task was measured for each pretraining objective. Combinations where the pretraining objective is the same as the downstream task were excluded.

The pretraining and finetuning is carried out on the same dataset. This evaluates the pretraining objectives as a means of extracting *additional information* from the same data, in contrast to evaluating their ability to benefit from additional data. Though pretraining on external data may prove to be effective, identifying a suitable pretraining dataset is challenging and this approach more directly evaluates the pretraining objectives.

5.2 Performance on Full Data

To first examine whether the pretraining objectives facilitate improved performance on downstream tasks a baseline model was trained for each down-

	BSP	DAP	NUR	NUG
	F-1	F-1	H@1	BLEU
None	**18.48**	40.33	63.72	14.21
NUR	17.80	43.25	–	15.39
NUG	17.96	42.31	**67.34**	–
MUR	16.76	**44.87**	62.38	15.27
InI	16.61	**44.84**	62.62	**15.52**

Table 1: Results of evaluating the chosen *pretraining objectives*, preceded by the baseline, on the four *downstream tasks*. This evaluation used all of the training data for the downstream tasks as described in Section 5.2.

stream task, using the entire set of MultiWoz data. The first row of Table 1 shows the performance of randomly initialized models for each downstream task. To evaluate the full capabilities of the pretraining objectives above, the pretrained models were used to initialize the models for the downstream tasks.

Results are shown on Table 1. This experimental setup speaks to the strength and the generality of the pretrained representations. Using unsupervised pretraining, the models produce dialog representations that are strong enough to improve downstream tasks. The learned representations demonstrate generality because the multiple downstream tasks benefit from the same pretraining. Rather than learning representations that are useful for just the pretraining objective, or for a single downstream task, the learned representations are general and beneficial for multiple tasks.

For the DAP and NUG downstream tasks, the pretrained models consistently outperformed the baseline. InI has the highest BLEU score for NUG. This may be a consequence of the importance of both global context representations and local utterance representations in sequence generation models. Both InI and MUR score much higher than the baseline and the other methods for DAP, which may be due to the fact that these two approaches are trained to learn a representation of each utterance rather than just an overall context representation. NUR has significant gains when pretraining with NUG, possibly because the information that must be captured to generate the next utterance is similar to the information needed to retrieve the next utterance. Unlike the other downstream tasks, BSP did not benefit from pretraining. A potential justification of this result is that due to the difficulty of the task, the model needs to resort

to word-level pattern matching. The generality of the pretrained representations precludes this.

5.3 Convergence Analysis

This experimental setup measures the impact of pretraining on the convergence of the downstream training. Sufficiently general pretraining objectives should learn to extract useful representations of the dialog context. Thus when fine-tuning on a given downstream task, the model should be able to use the representations it has already learned rather than having to learn to extract relevant features from scratch. The performance on all downstream tasks with the different pretraining objectives is evaluated at every epoch. The results are presented on Figure 1.

These figures show faster convergence across all downstream tasks with significant improvement over a random initialization baseline. The results show that performance on the initial epochs is considerably better with pretraining than without. In most cases, performance evens out during training, thus attaining results that are comparable to the pretraining methods on the full dataset. It is important to note that performance of the models after just a single epoch of training is significantly higher on all downstream tasks when the encoder has been pretrained. This underlines the usefulness of the features learned in pretraining.

The convergence of BSP shown in Figure 1 is very interesting. Though the baseline ultimately outperforms all other methods, the pretrained models attain their highest performance in the early epochs. This suggests that the representations learned in pretraining are indeed useful for this task despite the fact that they do not show improvement over the baseline.

5.4 Performance on Limited Data

Sufficiently strong and general pretrained representations, should continue to succeed in downstream evaluation even when fine-tuned on significantly less data. The performance on downstream tasks is evaluated with various amounts of fine-tuning data (1%, 2%, 5%, 10% and 50%).

The effect of the training data size for each downstream task is also evaluated. The performance of NUR with different amounts of training data is shown on Figure 2. With 5% of the fine-tuning data, the NUG pretrained model outperforms the baseline that used 10%. With 10%
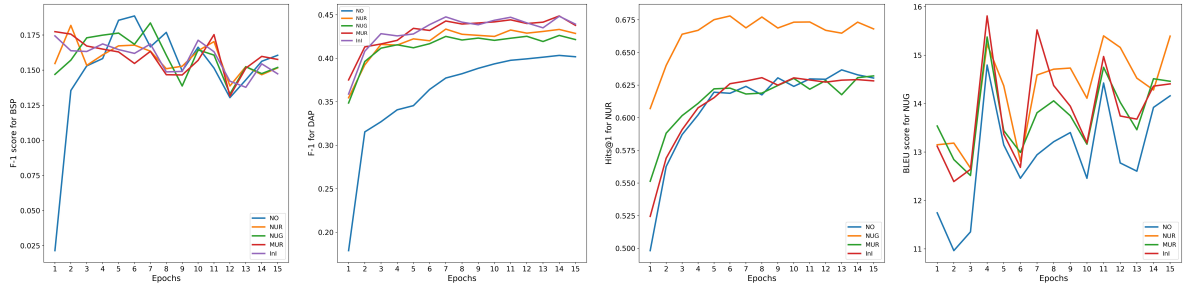
Figure 1: The performance of (from left to right) BSP, DAP, NUR, NUG across epochs with different pretraining objectives. For the BLEU-4 score in NUG, the results are noisy due to the metric being the BLEU score, however the general trend is still apparent.
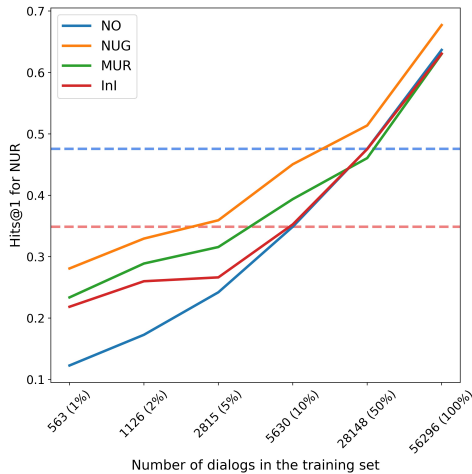

Figure 2: NUR Hits@1 at different training set sizes. The blue horizontal line is the baseline performance with 50% of the data. The red horizontal line is the baseline performance with 10% of the data.

of the fine-tuning data, this model outperforms the baseline that used 50% of the data.

Table 2 shows all of the results with 1% of the fine-tuning data, while Table 3 shows the results with 10% of the fine-tuning data. More results may be found in the Appendix ??.

	BSP	DAP	NUR	NUG
	F-1	F-1	H@1	BLEU
None	4.65	16.07	12.28	6.82
NUR	6.44	14.48	–	**11.29**
NUG	**7.63**	**17.41**	**28.08**	–
MUR	5.89	17.19	23.37	10.47
InI	6.18	12.20	21.84	11.10

Table 2: Performance using 1% of the data; the rows correspond to the pretraining objectives and the columns correspond to the downstream tasks.

The results shown here strongly highlight the effectiveness of pretraining. With a small fraction

	BSP	DAP	NUR	NUG
	F-1	F-1	H@1	BLEU
None	5.73	18.44	34.88	9.19
NUR	7.30	20.84	–	**14.04**
NUG	**9.62**	22.11	**45.05**	–
MUR	7.08	**22.24**	39.38	11.63
InI	7.30	20.73	35.26	13.23

Table 3: Results with 10% of the data; the rows correspond to the pretraining objectives and the columns correspond to the downstream tasks.

of the data, unsupervised pretraining shows competitive performance on downstream tasks.

When the amount of data is very limited, the best results were obtained by models pretrained with NUG. This may be indicative of the generality of NUG pretraining. Since the generation task is difficult, it is likely that the pretrained model learns to capture the most general context representation that it can. This makes the representations especially suitable for low resource conditions since NUG pretrained representations are general enough to adapt to different tasks given even very small amounts of data,

5.5 Domain Generalizability

Sufficiently general pretrained representations should facilitate domain generalizability on the downstream tasks, just as pretraining should encourage the downstream models to use domain agnostic representations and identify domain agnostic relationships in the data.

This experimental setup is designed to mimic the scenario of adding a new domain as the downstream task. It assumes that there are large quantities of unlabeled data for unsupervised pretraining in all domains but that there is a limited set of labeled data for the downstream tasks. More specif-

	BSP	DAP	NUR	NUG
	F-1	F-1	H@1	BLEU
None	4.07	15.22	13.62	7.80
NUR	**19.64**	17.88	–	9.97
NUG	17.11	**20.53**	**21.57**	–
MUR	15.84	17.45	21.06	9.81
InI	14.61	15.56	19.80	**10.87**

Table 4: Results of evaluating pretrained objectives on their capacity to generalize to the *restaurant* domain using only 50 in-domain samples and 2000 out-of-domain samples during training. The evaluation is carried out only on the in-domain test samples.

	< 3	$\geq 3 \ \& < 7$	≥ 7
None	11.02	14.17	15.30
NUR	**13.95**	15.08	15.88
MUR	12.21	15.36	16.10
InI	11.52	**15.40**	**16.63**

Table 5: Results on the downstream task of NUG, with different dialog context lengths ($<$ 3 utterances, 3-7 utterances, and $>$ 7 utterances.

ically, for each downstream task there are 1000 labeled out-of-domain examples (2% of the dataset) and only 50 labeled in-domain examples (0.1% of the dataset). The performance of the downstream models is computed only on the in-domain test samples, thereby evaluating the ability of our models to learn the downstream task on the limited in-domain data. The results on Table 4 show that pretraining produces more general representations and facilitates domain generalizability.

6 Discussion

The results with different experimental setups demonstrate the effectiveness of the pretraining objectives. Pretraining improves performance, leads to faster convergence, works well in low-data scenarios and facilitates domain generalizability. We now consider the respective strengths of the different pretraining objectives.

NUR and NUG are complementary tasks. Over all of the results, we can see that pretraining with either NUG or NUR, gives strong results when fine-tuning on the other one. This property, which has also been observed by Wolf et al. (2019), is a consequence of the similarity of the two tasks. Both for retrieval and generation, context encoding must contain all of the information that is necessary to produce the next utterance.

NUG learns representations that are very general. We see that NUG, especially in low data experiments, effectively transfers to many downstream tasks. This speaks to the generality of its representations. To auto-regressively generate the next utterance, the context encoder in NUG must capture a strong and expressive representation of the dialog context. This representation is all that the decoder uses to generate its response at word level so it must contain all of the relevant infor-

mation. Despite the similarity of NUG and NUR, generation is a more difficult task, due to the potential output space of the model. As such, the representations learned by NUG are more general and expressive. The representative capabilities of the encoder in a generation model are also demonstrated by the work of Adi et al. (2016).

InI and MUR learn strong local representations of each utterance. The two novel pretraining objectives, InI and MUR, consistently show strong improvement for the downstream NUG task. Both of these objectives learn local representations of each utterance in the dialog context since both of their respective loss functions use the representation of each utterance instead of just the final hidden state. In an effort to better understand the properties of the different objectives, Table 5 shows performance on the NUG task for different dialog context lengths.

Generating a response to a longer dialog context requires a strong local representation of each individual utterance. A model that does not capture strong representations of each utterance will likely perform poorly on longer contexts. For example, for a dialog in which the user requests a restaurant recommendation, in order to generate the system utterance that recommends a restaurant, the model must consider all of the past utterances in order to effectively generate the recommendation. If the local representations of each utterance are not strong, it would be difficult to generate the system output.

The results in Table 5 demonstrate that both InI and MUR strongly outperform other methods on long contexts, suggesting that these methods are effective for capturing strong representations of each utterance. Both MUR and InI perform poorly on shorter contexts. This further demonstrates that fine-tuned NUG models learn to rely on strong utterance representations, and therefore struggle when there are few utterances.

Using the same dataset for pretraining and

finetuning. The pretraining objectives demonstrate large improvements over directly training for the downstream task. No additional data is used for pretraining, which suggests that the proposed objective allow the model to extract stronger and more general context representations from the same data. The reduced data experiments show that pretraining on a larger corpora (i.e., the full data), results in strong performance on smaller task-specific datasets (i.e., the reduced data). As such, it is likely that pretraining on larger external data will result in further performance gains, however, it is challenging to identify a sufficient corpus.

7 Conclusion and Future Work

This paper proposes several methods of unsupervised pretraining for learning *strong* and *general* dialog context representations, and demonstrates their effectiveness in improving performance on downstream tasks with limited fine-tuning data as well as out-of-domain data. It proposes two novel pretraining objectives: *masked-utterance retrieval* and *inconsistency identification* which better capture both the utterance-level and context-level information. Evaluation of the learned representations on four downstream dialog tasks shows strong performance improvement over randomly initialized baselines.

In this paper, unsupervised pretraining has been shown to learn effective representations of dialog context, making this an important research direction for future dialog systems. These results open three future research directions. First, the models proposed here should be pretrained on larger external dialog datasets. Second, it would be interesting to test the representations learned using unsupervised pretraining on less-related downstream tasks such as sentiment analysis. Finally, the addition of word-level pretraining methods to improve the dialog context representations should be explored.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 861–872.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. 2017. Gated-attention readers for text comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1832–1846.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, et al. 2019. The second conversational intelligence challenge (convai2). *arXiv preprint arXiv:1902.00098*.

Ying Ding, Jianfei Yu, and Jing Jiang. 2017. Recurrent neural networks with auxiliary labels for cross-domain opinion target extraction. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 328–339.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Ryan Lowe, Nissan Pow, Iulian V Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 285.

Ryan Lowe, Iulian V Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. On the evaluation of dialogue systems with next utterance classification.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Ruslan Mitkov. 2014. *Anaphora resolution*. Routledge.

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.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 2227–2237.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 412–418.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*.

Marek Rei and Helen Yannakoudakis. 2017. Auxiliary objectives for neural error detection models. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 33–43.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, volume 16, pages 3776–3784.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Scott Thornbury and Diana Slade. 2006. *Conversation: From description to pedagogy*. Cambridge University Press.

Trieu Trinh, Andrew Dai, Thang Luong, and Quoc Le. 2018. Learning longer-term dependencies in rnns with auxiliary losses. In *International Conference on Machine Learning*, pages 4972–4981.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. Transfertransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149*.

Jianfei Yu and Jing Jiang. 2016. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 236–246.

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

Xiangyang Zhou, Daxiang Dong, Hua Wu, Shiqi Zhao, Dianhai Yu, Hao Tian, Xuan Liu, and Rui Yan. 2016. Multi-view response selection for human-computer conversation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 372–381.