ROBUST BELIEF STATE SPACE REPRESENTATION FOR STATISTICAL DIALOGUE MANAGERS USING DEEP AUTOENCODERS

Fotios Lygerakis[1], *Vassilios Diakoloulas*[1], *Michail Lagoudakis*[1], *Margarita Kotti*[2]

[1]School of Electrical & Computer Engineering, Technical University of Crete, Greece
[2]Speech Technology Group, Toshiba Research Cambridge, UK

ABSTRACT

Statistical Dialogue Systems (SDS) have proved their humongous potential over the past few years. However, the lack of efficient and robust representations of the belief state (BS) space refrains them from revealing their full potential. There is a great need for automatic BS representations, which will replace the old hand-crafted, variable-length ones. To tackle those problems, we introduce a novel use of Autoencoders (AEs). Our goal is to obtain a low-dimensional, fixed-length, and compact, yet robust representation of the BS space. We investigate the use of dense AE, Denoising AE (DAE) and Variational Denoising AE (VDAE), which we combine with GP-SARSA to learn dialogue policies in the PyDial toolkit. In this framework, the BS is normally represented in a relatively compact, but still redundant summary space which is obtained through a heuristic mapping of the original master space. We show that all the proposed AE-based representations consistently outperform the summary BS representation. Especially, as the Semantic Error Rate (SER) increases, the DAE/VDAE-based representations obtain state-of-the-art and sample efficient performance.

***Index Terms*—** denoising autoencoder, variational autoencoder, statistical dialogue system, dialogue manager, belief state representation

1. INTRODUCTION

In the past few years, there has been a rising interest in incorporating Spoken Dialogue Systems [1, 2, 3] in real-life applications. Many industries, like social robots [4], tourist information [5], banks [6], navigation [7], health care [8] and calling centers [9], have already incorporated Spoken Dialogue Systems to improve performance and offer better services. At its core, a Dialogue System utilizes a Dialogue Manager (DM) [10], which keeps track of the dialogue's state and predicts an action based on that state. So far, handcrafted DMs [11] have shown great performance in many domains, but their building complexity, cost of scalability, sensitivity to semantic errors, need for continuous supervision and domain-restricted usage have turned the research interest towards the development of Statistical DMs (SDMs) [10]. Their growth of popularity is a consequence of the numerous training corpora that became available online in the past few years, the fact that they enable the introduction of a continuous state space to the models and their generalization capability.

There are policy managers for the SDMs based on i) non-linear Reinforcement Learning algorithms, such as Deep Q-learning Networks (DQN) [12], Advantage Actor Critic (A2C) [13], episodic Natural Actor Critic (eNAC) [14], and ii) linear, like Gaussian Process SARSA (GP-SARSA) [15]. In [11] there is a comparison between those algorithms on different simulated environments and domains in which GP-SARSA seems to outperform all the other deep Neural Network (DNN) approaches. For the benchmarking procedure they use the summary BS (sumBS) representation [16, 17] to train the policy managers. The sumBS space is a heuristic mapping of the master (full) BS space. However, since it uses the domain ontology the sumBS vectors produced are domain-dependent, relatively high-dimensional, sparse and redundant. They also require domain expertise and they cannot be easily generalized and adapted to new domains.

Furthermore, in [11] the performance of the dialog manager degrades significantly in the presence of noise or when the domain complexity increases. This performance degradation could be attributed to the highly-dimensional and redundant sumBS vector. We believe that a more efficient BS representation could result in large performance gains particularly in difficult environments. In this paper, we investigate the use of dense AE, Denoising AE (DAE) and Variational Denoising AE (VDAE) to automatically extract robust and noise tolerant BS space representations.

1.1. Related Work

An alternative to the sumBS space representation is the Domain Independent Parameterization (DIP) [18]. DIP produces BS vectors using domain-independent dialogue features. They have therefore, a standard dimension for all domains and can be more compact than the sumBS vectors. However DIP mostly consists of hand-engineered features by the system designer which do not make use of domain-specific information and can still be noise sensitive, redundant and correlated. The BinLin/BinAux BS introduced in [19] is an-other alternative which considers limited information from the sumBS to build extremely compact BS representations. The main drawback of this featurization is that they need to be explicitly defined depending on the domain. Furthermore, they do not exhibit a stable performance under all conditions and they are sensitive to the input's uncertainty.

In recent work [20, 21], a feed forward network (FNN) and a recurrent neural network (RNN) are used to automatically train feature extractors in the form of feature functions for each slot to obtain dialogue BS abstractions. The training is performed jointly with a DQN-based policy. However, the resulting system is prone to non-optimal and slow convergence.

In [22, 23] feature selection is used to obtain lower-dimensional BS representation. However, most of the algorithms are tied to specific Reinforcement Learning algorithms such as LSPI, and cannot be generalized to other policy models. Furthermore, they do not exhibit robustness in the presence of noise.

1.2. Contribution

The contribution of this work is summarized as follows:
- Although AEs have been successfully used for non-linear feature extraction at several application domains, to the best of our knowledge, this is the first time that they are used to automatically obtain BS space representations in the framework of a SDS.
- We concentrate on noise-tolerant variations of the AEs such as the DAE and VDAE.
- We propose a novel training scheme which concurrently trains the AEs and performs policy optimization.

2. BACKGROUND AND METHODOLOGY

2.1. Dense Autoencoders

AEs are a family of NN topologies used for unsupervised learning which have been successfully employed for non-linear feature extraction in several application domains [24, 25, 26, 27]. Due to their architecture, AEs are forced to learn lower dimensional and more robust representations of the input vector $\boldsymbol{x}$. Typically, AEs can be seen as the concatenation of two networks. The first one is the encoder, which projects the input layer to a lower-dimensional latent space. The second one is the decoder, which takes the encoded compact representation and projects it back to the original input. Their architecture exhibits a perfect symmetry with regards to the central hidden layer which is the lowest-dimensional one and serves as the coding layer.

In this paper, the AEs are build as deep, feed-forward and dense NNs. The input $\boldsymbol{x} \in \mathbb{R}^{d_x \times 1}$ denotes the sumBS vector which has a variable length d_x depending on the chosen domain. The input layer is then projected to the first hidden layer $h_1(\boldsymbol{x}) \in \mathbb{R}^{d_{h1} \times 1}$ through a typical mapping of the form:

$$h_1(\boldsymbol{x}) = \alpha\left(W_{h_1 x} * \boldsymbol{x}\right) \tag{1}$$

where $\alpha()$ denotes the activation function, here $tanh$, and $W_{h_1 x} \in \mathbb{R}^{d_1 \times d_x}$ defines the matrix of weights connecting the input and the first hidden layer. Several hidden layers $j \in 1, \cdots, \lceil \frac{N}{2} \rceil$, can then be added with gradually reduced dimensions until the central, bottleneck layer $j = \lceil \frac{N}{2} \rceil$, which serves as the output of the encoder. N denotes the total number of hidden layers of the AE. The corresponding hidden representation of any hidden layer j is then defined as a mapping from the previous layer $j-1$ as follows:

$$h_j(\boldsymbol{x}) = \alpha\left(W_{h_j h_{j-1}} * h_{j-1}(\boldsymbol{x})\right) \tag{2}$$

The latent representation from the bottleneck layer $\boldsymbol{z} = h_{\lceil \frac{N}{2} \rceil}(\boldsymbol{x})$ is then mapped back to the input $\boldsymbol{x}$ through a reverse mapping function using the corresponding weight matrices of the decoder W^{dec}. In this work, the decoder's weight matrices are tied to the corresponding encoder's weight matrices $W^{dec} = W^T$.

2.2. Denoising Autoencoders

DAEs [28] are essentially dense AEs which are trained to represent the corrupted or noisy input vectors, providing robustness. In a typical SDS, noise can be introduced due to a multitude of factors, including the errors of the recognizer, the semantic errors (e.g. acoustic confusability, ambiguity of natural language, incomplete utterances, etc.), as well as the uncertainty of user's goal. In specific, the input of a DAE is a corrupted version $\widetilde{\boldsymbol{x}}$ of the clean sumBS vector $\boldsymbol{x}$:

$$\widetilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{n} \tag{3}$$

where $\boldsymbol{x}$ is the target in the output layer, and $\boldsymbol{n}$ denotes the noise vector, produced by an unknown distribution, which has the same size as the clean sumBS $\boldsymbol{x}$. Noise $\boldsymbol{n}$ is added to $\boldsymbol{x}$ artificially using the SER percentage as probability P_{SER}. Specifically, the corrupted vector $\widetilde{\boldsymbol{x}}$ is obtained by keeping, with probability $1 - P_{SER}$, the true semantic information for a slot, and making random selections from all the available values in the ontology, with probability P_{SER}. It is worth noting that this artificial corruption process can be applied to data from both real and simulated dialogues.

Using an adequately large number of $[\boldsymbol{x}, \widetilde{\boldsymbol{x}}]$ sample pairs, a DAE can be trained to approximate the noise distribution and appropriately filter out the noise. The parameters of the DAE are trained so as to minimize the average reconstruction error J defined as the mean squared error:

$$J(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{E}\left[||\boldsymbol{x} - \boldsymbol{y}||^2\right] \tag{4}$$

where $\boldsymbol{y}$ is the prediction obtained at the output of the network's decoder when we input the noisy $\widetilde{\boldsymbol{x}}$, and $\boldsymbol{x}$ is the original non-corrupted vector.

2.3. Variational Denoising Autoencoders

The AE and DAE networks may suffer from poor generalization to unseen data, particularly considering the sparse and continuous nature of the sumBS vector in the input. This is mainly attributed to the form of the latent space in which the training data are projected, which may contain gaps corresponding to forms of training samples that were never used as input. The generation of latent representations for these unseen data forms could be very poor, influencing the accuracy of the system.

To overcome this problem, we investigate the use of Variational AEs (VAEs) [29, 30]. VAEs create continuous latent spaces, since the latent vectors $\boldsymbol{z}$ are generated randomly from a parametric inference model defined as a multivariate Gaussian distribution $q(\boldsymbol{z}|\boldsymbol{x} : \boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{z} : \boldsymbol{\mu}, \boldsymbol{\sigma})$, where $\boldsymbol{\lambda} = \{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$, $\boldsymbol{\mu}$ is the mean vector of the Gaussian and $\boldsymbol{\sigma}$ is the corresponding vector of standard deviations. The $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ vectors are approximated through a corresponding pair of weight matrices $W_{\boldsymbol{\mu}}$ and $W_{\boldsymbol{\sigma}}$ in the bottleneck layer $h_{\lceil \frac{N}{2} \rceil}$, which are optimized as parameters of the neural network.

Optimization of the VAE is made on the basis of the following loss function:

$$J_{VAE}(\boldsymbol{x}, \boldsymbol{y}) = J(\boldsymbol{x}, \boldsymbol{y}) + D_{KL}\left(q(\boldsymbol{z}|\boldsymbol{x} : \boldsymbol{\lambda}) || p(\boldsymbol{z})\right) \quad (5)$$

where D_{KL} denotes the KL-Divergence among the true latent variable distribution $p(\boldsymbol{z})$ which is typically chosen to follow the standard Gaussian $\mathcal{N}(0, 1)$ and the approximation $q(\boldsymbol{z}|\boldsymbol{x} : \boldsymbol{\lambda})$ learned by the encoder.

In the proposed Variational DAE (VDAE) scheme, the inference model $q(\boldsymbol{z}|\tilde{\boldsymbol{x}} : \boldsymbol{\lambda})$ learns to generate latent vectors from corrupted sumBS state vectors $\tilde{\boldsymbol{x}}$ that have been injected with noise using the method described in Section 2.2. In this way, the VDAE will be trained to be tolerant in noise.

2.4. Concurrent Training

In this work, both the policy manager and the parameters of the AEs are optimized in parallel using batches of dialogue episodes. The AEs take as input sumBS vectors $\boldsymbol{x}$, while the policy manager takes as input the AE's bottleneck layer $h_{\lceil \frac{N}{2} \rceil}(\boldsymbol{x})$. To speed up the convergence of this parallel process we perform pre-training on the AEs based on randomly generated sumBS vectors. The pre-training process is terminated when the loss function drops below a threshold. Further optimization is then performed on-line to dynamically adapt the models to the environment's uncertainty. This process is summarized in Algorithm 1.

Algorithm 1 Concurrent Training Scheme

AE.preTrain()
for $iteration \in (1, NumOfDialogueBatches)$ **do**
    for $Dialogue$ in $DialogueBatch$ **do**
        DialSimulation.Start()
        while $!(DialSimulation.Ended()) = True$ **do**
            Policy.saveToBatch(Episode)
            AE.saveToBatch(Episode)
            if $AE.BatchisFull() = True$ **then**
                AE.train()
    Policy.train()
    Policy.evaluate()

3. EXPERIMENTAL SETUP

For our experiments, we used the GP-SARSA policy algorithm in the PyDial framework on simulated standard users [31]. The choice of GP-SARSA was made upon its outstanding performance in comparison to other approaches [11]. We evaluated our algorithms in the following three domains:
- *Cambridge Restaurants (CR)*, which is the most common domain in the literature. It produces a **268**-dimensional sumBS vector.
- *San Francisco Restaurants (SFR)*, which has a higher-dimensional (**636**-dimensions) sumBS vector.
- *Laptops11 (LAP11)*, which is among the most difficult domains, especially for higher SER. LAP11 has a **257**-dimensional sumBS vector.

For each domain we defined a different domain-specific AE/DAE/VDAE network. Although the input and the output layer of these networks had a different dimension for each domain, the rest of the network topology shared exactly the same characteristics for all the domains. Particularly, we considered networks of 5 fixed-size hidden layers $h1 : 200$, $h2 : 100$, $h3 : 50$, $h4 : 100$, $h5 : 200$, where the number indicates the hidden nodes in each layer. We also examined a deeper 7 hidden layer architecture with hidden layers dimensions $h1 : 200$, $h2 : 100$, $h3 : 50$, $h4 : 30$ $h5 : 50$ $h6 : 100$, $h7 : 200$. The latent space of the bottleneck layer which also defines the dimension of the BS space representation is therefore 50 or 30 respectively.

Furthermore, they shared the same hyperparameters and optimization methods. For instance, we used ADAM optimizer and we applied dropout in the encoder with a dropout rate of 0.6 to avoid over-fitting. The learning rate (LR) was configured as time-based, exponential decaying and the initial value was set to 10^{-4} with a decay parameter of 10^{-2}. Offline training was performed in 10 batches of 300 dialogues each and we evaluated each batch on 300 test dialogues.

3.1. Experimental Results

Although we used both the shallower and deeper topologies for all of our AE variations, we only present results concerning the best performing topology, that is, the 5 hidden layer for the dense AE and the deeper, 7 hidden layer one, for each of the DAE and VDAE. To obtain fair results, we performed

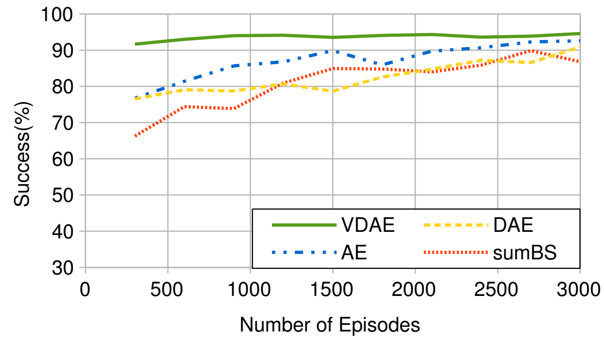
Episodes: 3000		Domains		
SER	BS	CR	LAP11	SFR
0%	sumBS	98.4%(±1.1)	86.8%(±3.9)	95.2%(±1.3)
	AE	**99.3%**(±0.5)	92.6%(±1.9)	95.3%(±0.8)
	DAE	97.3%(±0.8)	90.9%(±1.7)	**95.4%**(±2.1)
	VDAE	96.5%(±2.7)	**94.7%**(±1.7)	93.7%(±2.1)
15%	sumBS	96.4%(±2.2)	66.5% (±2.3)	81.6% (±1.6)
	AE	**96.5%**(±1.1)	68.9%(±10.1)	89.3%(±1.8)
	DAE	91.9%(±2.9)	89.7%(±3.1)	**95.1 %**(±1.4)
	VDAE	95.5%(±0.8)	**91.7%**(±0.5)	93.6%(±1.0)
30%	sumBS	88.5%(±4.1)	51.4%(±9.3)	66.3%(±5.3)
	AE	92.2%(±1.1)	50.1%(±10.3)	69.4%(±2.3)
	DAE	**92.9%**(±2.4)	84.3%(±4.0)	**94.9%**(±1.28)
	VDAE	92.9%(±1.3)	**90.2%**(±1.8)	89.9%(±2.7)
45%	sumBS	78.0%(±3.4)	24.1%(±5.5)	53.9%(±6.8)
	AE	78.1%(±3.2)	38.7%(±5.3)	36.9%(±7.9)
	DAE	91.2%(±5.4)	**88.0%**(±2.9)	81.9 %(±7.8)
	VDAE	**92.3%**(±3.3)	87.9%(±2.7)	**88.6%**(±2.9)

Table 1: Average success comparison with the sumBS baseline for 3000 dialogue episodes. Standard deviation in parenthesis. Best score in bold.

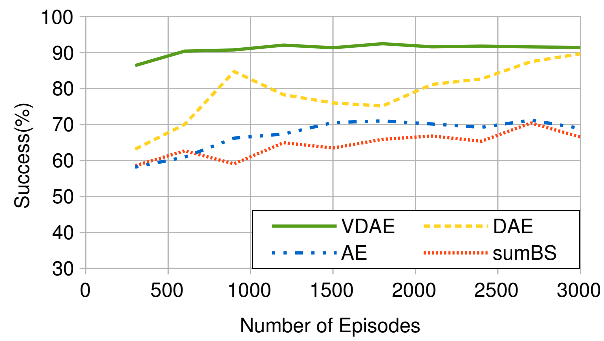
five independent runs of the same experiment using different initialization of the random generator and we calculated average dialogue success rates. Table 1 summarizes the performance for all the domains and BS representations. We used four different levels of SER (0%, 15%, 30% and 45%), consistently in both training and evaluation.

It can be seen that the dialogue manager benefits from the AE-based representation for 0% SER, but is unable to provide noise-robust features for higher SERs. On the other hand, as the presence of noise increases, the representation based on DAEs and VDAEs shows great potential. Specifically, they both maintain a performance close to 90% even for noise levels as high as 45%. Their dominance in performance is even more apparent in the difficult domains of LAP11 and SFR [11] , where the performance of sumBS and dense AEs dramatically degrades . Although we only show the performance of the best performing 7 hidden layer DAE/VDAE architecture, it is important to note that even their shallower 5 hidden layer topologies maintain a performance only $6\% - 12\%$ lower than the deeper ones and they still dramatically outperform the baseline representations of sumBS and AE.

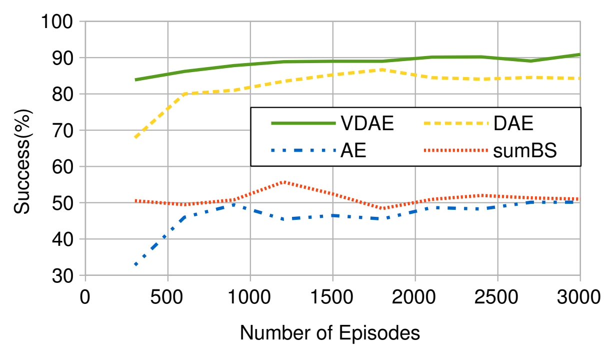
The diagrams in Figures 1 and 2a show the evolution of the average dialogue success rates in the LAP11 domain, which is the most difficult one, for different SER levels and the different BS representations considered in the paper. It can be seen that in all these diagrams, the policy shows a relatively smooth and fast convergence for all the AE-based representations. However, the performance of the VDAE clearly excels not only as the best performer, but also as the representation which achieves high performance even with as little as 300 episodes. Although there are similar findings for the remaining domains considered, we choose to present in



(a) LAP11 0%.



(b) LAP11 15%.



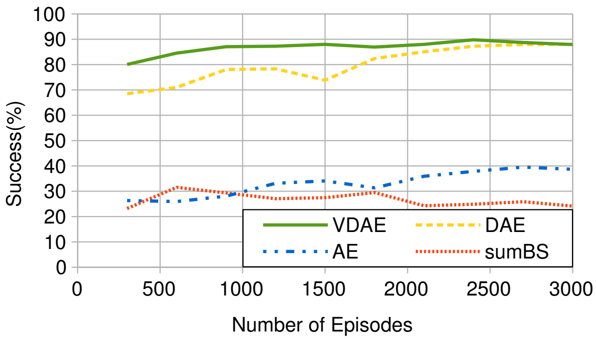
(c) LAP11 30%.

Fig. 1: LAP11 Domain in Multiple Environments

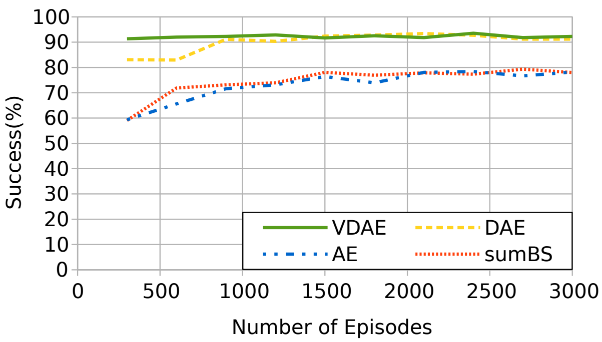
Figure 2 the corresponding performance only in the extremely noisy environment (45% SER) across all the domains. These diagrams demonstrate the consistent behaviour of the VDAE among all the domains since in all cases it achieves the highest accuracy with the smallest number of training episodes.

3.2. Discussion

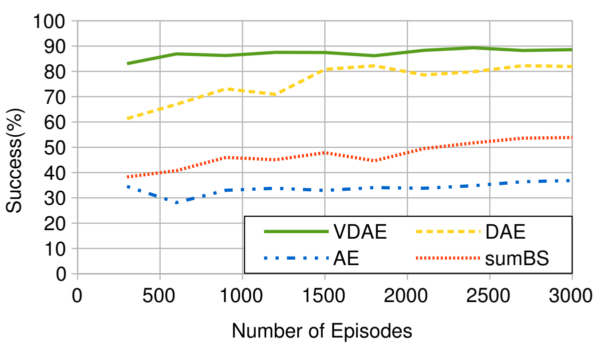
A direct comparison with the state-of-the-art is not feasible due to the different experimental protocols, domains, and



(a) LAP11 45%.



(b) CR 45%.



(c) SFR 45%.

Fig. 2: LAP11, CR and SFR domains in 45% SER

policies of previous works. For example, in [18], a first version of the DIPs in conjunction with GP-SARSA is tested on both the CR and a precursor of the LAP11 domain. After training on 1000 dialogues with 15% SER, the reported accuracy is approximately 95% for the CR, which is comparable to the average accuracy of 94.6% that we obtain using dense AEs and lower than the accuracy of 96.2% of the VDAE in the same environment and domain after 900 dialogues. In [19], where the CR domain is studied, the examined BS representation is DIPs, BinLin, and BinAux and the policy manager is

SER/ ENV	**Model**	CR		LAP11		SFR	
		Succ	Rew	Succ	Rew	Succ	Rew
0%	DAE	**97.3%**	**13.1**	90.9%	**10.6**	**95.4%**	**11.9**
	VDAE	96.5%	11.3	**94.7%**	8.1	93.7%	7.8
Env.1	Feudal-DQN	89.3%	11.7	65.5%	5.7	71.1%	7.1
15%	DAE	91.9%	**11.7**	89.7%	**10.1**	**95.1%**	**11.5**
	VDAE	**95.5%**	10.3	**91.7%**	6.6	93.6%	7.5
Env.3	Feudal-DQN	92.6%	**11.7**	89.6%	9.4	90.0%	9.7
30%	DAE	**93.0%**	**11.9**	84.3%	**9.0**	**94.9%**	**11.2**
	VDAE	92.9%	9.5	**90.2%**	6.0	89.9%	5.9
Env.6	Feudal-DQN	90.6%	10.4	78.5%	6.0	83.0%	7.1

Table 2: DAE and VDAE average success and reward (3000 episodes) compared to the Feudal-DQN system [21] (4000 episodes)

A2C, whereas the number of training dialogues is 2000. The reported accuracy ranges from 17% to 35% lower compared to our VDAE system, depending on the environment.

In [20, 21] DIP, FFN and RNN Feudal features are exploited. The policies considered are NN-based and the results presented are for 4000 episodes. Nevertheless, since they consider the same domains and matching conditions for the environments 1, 3 and 6 which correspond to 0%, 15% and 30% respectively, we show in Table 2 a comparison of both the average success and the reward. It can be seen that the proposed VDAE with GP-SARSA dominates in all cases in terms of success rate, although it is trained on less episodes. DAE on the other hand, obtains the highest reward.

It is worth noting that since the AE/DAE/VDAE topology in our proposed scheme is domain-independent with the exception of the input and output layer, new AEs could be rapidly optimized in a transfer learning framework [32, 33]. This is particularly useful when new entries in a domain's ontology are introduced, as well as for optimizing new AEs for new domains with limited data, making our approach domain-transferable.

4. CONCLUSIONS

In this paper, we propose a novel use of AEs, DAEs, and VDAEs for representing the BS for SDMs. Our motivation is to create low-dimensional, fixed-size, and robust representations which are automatically extracted from a set of training dialogues. Using corrupted training data we obtain representations tolerant in SER noise. Furthermore, the learned representations are trained concurrently with the policy manager in a novel training scheme. The experimental results show that the proposed scheme is very efficient, outperforming significantly, even with limited training samples, the baseline sumBS, and other state-of-the-art approaches for different domains and noise levels. The highest gain refers to environments with 45% SER and difficult domains such as the demanding Laptops11, where an absolute improvement of 63.79% is achieved, confirming the proposed methods ability.

5. REFERENCES

[1] Nicholas Roy, Joellea Pineau, and Sebastian Thrun, "Spoken dialogue management using probabilistic reasoning," in *ACL*, 2000.

[2] Steve J. Young, "Talking to machines (statistically speaking)," in *INTERSPEECH*, 2002.

[3] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.*, vol. 101, pp. 99–134, 1998.

[4] Ridong Jiang, Yeow Tan, Dilip Kumar Limbu, Tran Anh Dung, and Haizhu Li, "Component pluggable framework and its application to social robots," in *Natural Interacton with Robots and Smartphones, Putting Spoken Dialog Systems into Practice*, 2014.

[5] Pawel Budzianowski, Stefan Ultes, Pei-Hao Su, Nikola Mrkšić, Tsung-Hsien Wen, Iñigo Casanueva, Lina Maria Rojas-Barahona, and Milica Gašić, "Subdomain modelling for dialogue management with hierarchical reinforcement learning," in *SIGDIAL*, 2017.

[6] Håkan Melin, Anna C Sandell, and Magnus Ihse, "Cttbank: A speech controlled telephone banking system - an initial evaluation," in *Speech Music and Hearing Quarterly Progress and Status Report (TMH-QPSR)*, 2001, vol. 1.

[7] Kyungduk Kim, Cheongjae Lee, Sangkeun Jung, and Gary Geunbae Lee, "A frame-based probabilistic framework for spoken dialog management using dialog examples," in *SIGDIAL*, 2008.

[8] Alexandros Papangelis, Robert Gatchel, Vangelis Metsis, and Fillia Makedon, "An adaptive dialogue system for assessing post traumatic stress disorder," in *PETRA*, 2013.

[9] Hilda Hardy, Tomek Strzalkowski, and Min Wu, "Dialogue management for an automated multilingual call center," in *HLT-NAACL*, 2003.

[10] Steve J. Young, Milica Gašić, Blaise Thomson, and Jason D. Williams, "POMDP-based statistical spoken dialog systems: A review," *Proc. IEEE*, vol. 101, pp. 1160–1179, 2013.

[11] Iñigo Casanueva, Pawel Budzianowski, Pei-Hao Su, Nikola Mrkšić, Tsung-Hsien Wen, Stefan Ultes, Lina Maria Rojas-Barahona, Steve J. Young, and Milica Gašić, "A benchmarking environment for reinforcement learning based task oriented dialogue management," *CoRR*, vol. abs/1711.11023, 2017.

[12] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller, "Playing atari with deep reinforcement learning," *CoRR*, vol. abs/1312.5602, 2013.

[13] Mehdi Fatemi, Layla El Asri, Hannes Schulz, Jing He, and Kaheer Suleman, "Policy networks with two-stage training for dialogue systems," in *SIGDIAL*, 2016.

[14] Pei-Hao Su, Pawel Budzianowski, Stefan Ultes, Milica Gašić, and Steve J. Young, "Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management," in *SIGDIAL*, 2017.

[15] Milica Gasic and Steve J. Young, "Gaussian processes for pomdp-based dialogue manager optimization," *IEEE/ACM Trans. Audio, Speech, and Language Processing*, vol. 22, pp. 28–40, 2014.

[16] J. D. Williams and S. Young, "Scaling up pomdps for dialog management: The "summary pomdp" method," in *IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.*, 2005, pp. 177–182.

[17] Jason Williams and Steve Young, "Scaling pomdps for spoken dialog management," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, pp. 2116 – 2129, 2007.

[18] Zhuoran Wang, Tsung-Hsien Wen, Pei-Hao Su, and Yannis Stylianou, "Learning domain-independent dialogue policies via ontology parameterisation," in *SIGDIAL*, 2015.

[19] Margarita Kotti, Vassilios Diakoloukas, Alexandros Papangelis, Michail Lagoudakis, and Yannis Stylianou, "A case study on the importance of belief state representation for dialogue policy management," *INTERSPEECH*, vol. 986-990, 2018.

[20] Iñigo Casanueva, Paweł Budzianowski, Stefan Ultes, Florian Kreyssig, Bo-Hsiang Tseng, Yen-Chen Wu, and Milica Gašić, "Feudal dialogue management with jointly learned feature extractors," in *SIGDIAL*, 2018.

[21] Iñigo Casanueva, Paweł Budzianowski, Pei-Hao Su, Stefan Ultes, Lina M. Rojas Barahona, Bo-Hsiang Tseng, and Milica Gašić, "Feudal reinforcement learning for dialogue management in large domains," in *NAACL-HLT*, 2018, pp. "714–719".

[22] Lihong Li, Jason D. Williams, and Suhrid Balakrishnan, "Reinforcement learning for dialog management using least-squares policy iteration and fast feature selection," in *INTERSPEECH*, 2009.

[23] De-Rong Liu, Hong-Liang Li, and LiDing Wang, "Feature selection and feature learning for high-dimensional

batch reinforcement learning: A survey," *International Journal of Automation and Computing*, vol. 12, pp. 229–242, 2015.

[24] Harris Partaourides and Sotirios P. Chatzis, "Asymmetric deep generative models," *Neurocomputing*, vol. 241, pp. 90–96, 2017.

[25] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, 2010.

[26] Alex Krizhevsky and Geoffrey E Hinton, "Using very deep autoencoders for content-based image retrieval.," in *ESANN*, 2011.

[27] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori, "Speech enhancement based on deep denoising autoencoder.," in *INTERSPEECH*, 2013, pp. 436–440.

[28] Pascal Vincent, Isabelle Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn.Res*, pp. 3371–3408, 2010.

[29] Diederik P. Kingma and Max Welling, "Auto-encoding variational bayes," *CoRR*, 2014.

[30] Kuan Han, Haiguang Wen, Junxing Shi, Kun-Han Lu, Yizhen Zhang, and Zhongming Liu, "Variational autoencoder: An unsupervised model for modeling and decoding fmri activity in visual cortex," *bioRxiv*, 2018.

[31] Stefan Ultes, Lina Maria Rojas-Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Iñigo Casanueva, Pawel Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gašić, and Steve J. Young, "PyDial: A multi-domain statistical dialogue system toolkit," in *ACL*, 2017.

[32] C. Kandaswamy, L. M. Silva, L. A. Alexandre, R. Sousa, J. M. Santos, and J. M. de S, "Improving transfer learning accuracy by reusing stacked denoising autoencoders," in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2014, pp. 1380–1387.

[33] Fuzhen Zhuang, Xiaohu Cheng, Ping Luo, Sinno Jialin Pan, and Qing He, "Supervised representation learning: Transfer learning with deep autoencoders," in *Proceedings of the 24th International Conference on Artificial Intelligence*, 2015, pp. 4119–4125.