INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Deep Stacked Autoencoders for Spoken Language Understanding

Killian Janod[1,2], *Mohamed Morchid*[1], *Richard Dufour*[1], *Georges Linarès*[1], *Renato De Mori*[1,3]

[1]LIA - University of Avignon (France)
[2]ORKIS - Aix en Provence (France)
[3]McGill University - Montreal, Quebec (Canada)
kjanod@orkis.com, firstname.lastname@univ-avignon.fr

Abstract

The automatic transcription process of spoken document results in several word errors, especially when very noisy conditions are encountered. Document representations based on neural embedding frameworks have recently shown significant improvements in different Spoken and Natural Language Understanding tasks such as denoising and filtering. Nonetheless, these methods mainly need clean representations, failing to properly remove noise contained in noisy representations. This paper proposes to study the impact of residual noise contained into automatic transcripts of spoken dialogues in highly abstract spaces from deep neural networks. The paper makes the assumption that the noise learned from "clean" manual transcripts of spoken documents moves down dramatically the performance of theme identification systems in noisy conditions. The proposed deep neural network takes, as input and output, highly imperfect transcripts from spoken dialogues to improve the robustness of the document representation in a noisy environment. Results obtained on the DECODA theme classification task of dialogues reach an accuracy of 82% with a significant gain of about 5%.

Index Terms: spoken dialogues, deep neural networks, denoising autoencoders, deep stacked autoencoders.

1. Introduction

Research on spoken language understanding (SLU) items such as conversation analysis, speech analytics, topic identification and segmentation are receiving an increasing attention as documented in [1, 2, 3, 4, 5] and [6] respectively.
In this paper, another original solution based on Deep Stacked Autoencoders (DSAEs) [7] is proposed. These DSAEs are trained to extract latent features robust to the noise affecting corrupted input. This robust representation can be extended to topic identification of any type of possibly corrupted or partially corrupted documents. Experimental evidence is provided that, using a multilayer perceptron (MLP) classifier fed by DSAEs features, provide higher theme identification accuracy than the same classifier fed by other types of features estimated for reconstructing the manually or automatically transcribed input. Denoising Autoencoders (DAE) [8] have been recently generalized into Generative Stochastic Networks (GSN) [7] with the purpose of learning estimations of the input data distribution. The DSAEs proposed in this paper have the purpose of modifying the latent representation of a noisy input distribution to approach the classification accuracy that would be obtained with a clean representation of the same document. Such a denoising task also differs from adapting a classification process to improve the classification accuracy of the same document. The rest of this paper is organized as follows. Related work and proposed approaches are described in Sections 2 and 3. Section 4 presents the experimental protocol while Section 5 reports results. Finally, Sections 6 and 7 discuss and conclude the work.

This work was funded by the ANR-14-CE24-0022 GaFes project supported by the French National Research Agency.

2. Related Work and Motivations

Different types of multilayer networks have been proposed for denoising purposes. Among them, [9] and [10] propose an autoassociative memory to retrieve data from partial information. More recently, solutions have been proposed by [8, 11, 12]. Recent advances in deep learning [7] have shown impressive performance in image classification and regression [9, 10], language [8, 11] or speech [12, 13] processing. Autoencoders [10] are widely used to obtain latent representations capturing sufficient information to reconstruct the input data. These representations are effectively used for pre-training deep neural networks (DNN) [14]. Deep Autoencoders (DAEs) have been proposed [15] to improve the reconstruction robustness in presence of noise affecting input data. Interesting results have been obtained with DAEs in domains such as medicine [16], biology [17], image processing [18], motions [19], music [20] and speech [21]. These DAEs use the same vector for representing inputs and outputs. Their parameters are estimated by artificially corrupting the input with additive noise. There is no evidence that DAEs with a single hidden layer effectively capture intra- and inter-words distribution structures.
The proposed deep supervised autoencoder extracts from homogeneous input/output vectors a robust representation in a small latent subspace. This work differs from the previous ones because the "noise" part of the input vector is obtained with predefined transformation function (gaussian noise…). Moreover, the proposed Deep Stacked Autoencoder (DSAE) learns robust features from highly imperfect transcripts obtained from an ASR system in actual use conditions.

3. Proposed approach

This section reviews basic AE and DAE concepts to highlight the ideas that inspired the proposed approach and the novelties that have been introduced to properly address the problem of identifying the themes of real-life telephone dialogues.

3.1. Document representation

The task considered in this paper is the reconstruction of a feature distribution corrupted by the imprecision of the feature ex-

http://dx.doi.org/10.21437/Interspeech.2016-63

traction component. Features are obtained from a set of 707 content words exhibiting high mutual information with the application domain themes. Given a document of a corpus D, an input feature vector $\mathbf{x}$ is defined with the i-th element $\mathbf{x}_i$ computed as follows:

$$\mathbf{x}_i = |t_i| \times \Delta(t_i) \quad (1)$$

where $|t_i|$ is the number of occurrences of ti in the document and $\Delta(t_i)$ is the product between the inverse document frequency and the word purity defined with the Gini criterion.

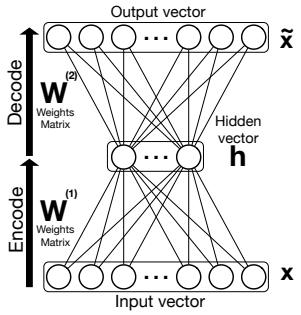

Figure 1: Illustration of the autoencoder model with an input features layer, one hidden layer and the output units. For readability, biases are omitted here.

3.2. Basic Concepts of an Autoencoder (AE)

The autoencoder (AE) network is a feed-forward three-layered neural network made of an encoder and a decoder (see Figure 1). The encoder computes a hidden representation of $\mathbf{x}$ made of a vector $\mathbf{h}$ of size m (number of hidden units) as follows:

$$\mathbf{h} = \sigma(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) \,, \quad (2)$$

where $\mathbf{W}^{(1)}$ is a $m \times n$ weight matrix and $\mathbf{b}^{(1)}$ is a m-dimensional bias vector. $\sigma(.)$ is the hyperbolic tangent activation function defined as:

$$\sigma(\mathbf{y}) = \frac{e^{\mathbf{y}} - e^{-\mathbf{y}}}{e^{\mathbf{y}} + e^{-\mathbf{y}}} \quad (3)$$

The decoder attempts to reconstruct the input vector $\mathbf{x}$ from the hidden vector $\mathbf{h}$ to obtain the output vector $\tilde{\mathbf{x}}$:

$$\tilde{\mathbf{x}} = \sigma(\mathbf{W}^{(2)}\mathbf{h} + \mathbf{b}^{(2)}) \,, \quad (4)$$

where the reconstructed vector $\tilde{\mathbf{x}}$ is a n-dimensional vector, $\mathbf{W}^{(2)}$ is a $n \times m$ weight matrix and $\mathbf{b}^{(2)}$ is a n-dimensional bias vector.

During learning, the autoencoder attempts to reduce the reconstruction error l between $\mathbf{x}$ and $\tilde{\mathbf{x}}$ by using the traditional Mean Square Error (MSE) [22] ($l_{\text{MSE}}(\mathbf{x}, \tilde{\mathbf{x}}) = ||\mathbf{x} - \tilde{\mathbf{x}}||^2$) for minimizing the total reconstruction error L_{MSE} with respect to the parameters set $\theta = \{\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(2)}\}$:

$$L_{\text{MSE}}(\theta) = \frac{1}{d}\sum_{\mathbf{x}\in D} l_{\text{MSE}}(\mathbf{x}, \tilde{\mathbf{x}}) = \frac{1}{d}\sum_{\mathbf{x}\in D} ||\mathbf{x} - \tilde{\mathbf{x}}||^2 \quad (5)$$

Two autoencoders are trained in this paper. One for reconstructing features ($\mathbf{x}^{(\text{ASR})}$) of automatic ASR transcriptions and the other for recounting features ($\mathbf{x}^{(\text{TRS})}$) of manually transcribed documents. The parameters estimated for these autoencoders are used to initialize the elements of the weight matrices $\mathbf{W}^{(\text{ASR})}$ and $\mathbf{W}^{(\text{TRS})}$ used to estimate the bottleneck features of the denoising autoencoder that will be introduced later on.

3.3. Denoising Autoencoder (DAE)

The aim of an autoencoder is to build a robust generative model to encode and decode a given input vector $\mathbf{x}$ in a latent space $\mathbf{h}$. During the learning process, the autoencoder fails to separate robust features relevant information and residual noise for a given input distribution [23]. For this reason, [23] proposes to corrupt the inputs before the encoding process and then decode this noisy representation to a clean one with a Denoising Autoencoder (DAE). In this way, the DAE is expected to recover a clean representation from a noisy input by learning a robust generative model.

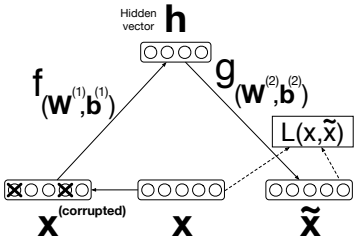

Figure 2: Denoising autoencoder architecture. An input vector $\mathbf{x}$ is stochastically corrupted to obtain $\mathbf{x}^{(\text{corrupted})}$, then mapped to the latent space $\mathbf{h}$ to extract the reconstructed vector $\tilde{\mathbf{x}}$. The reconstruction error is evaluated with the loss function L.

Figure 2 shows the scheme of a denoising autoencoder architecture. In this model, the input vectors $\mathbf{x}$ are considered as "clean" representations. The aim of this DAE is to obtain a robust reconstruction from an input vector to a clean output one. Therefore, $\mathbf{x}$ is artificially corrupted via a function that can be an Additive isotropic *Gaussian Noise* (GS) $\mathbf{x}^{(\text{corrupted})}|\mathbf{x} \sim \mathcal{N}\left(\mathbf{x}, \sigma^2\mathbf{I}\right)$ or a *Salt-and-pepper Noise* (SN). This corrupted input $\mathbf{x}^{(\text{corrupted})}$ is then mapped to a hidden layer $\mathbf{h} = f_{(\mathbf{W}^{(1)},\mathbf{b}^{(1)})} = \sigma(\mathbf{W}^{(1)}\mathbf{x}^{(\text{corrupted})} + \mathbf{b}^{(1)})$. The reconstructed vector $\tilde{\mathbf{x}}$ is obtained in a same manner $\tilde{\mathbf{x}} = g_{(\mathbf{W}^{(2)},\mathbf{b}^{(2)})} = \sigma(\mathbf{W}^{(2)}\mathbf{h} + \mathbf{b}^{(2)})$. During the learning process, the denoising autoencoder learns the parameters $\theta = \{\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(2)}\}$ to minimize the reconstruction error $L(\mathbf{x}, \tilde{\mathbf{x}})$. The motivation for using this type of DAE is that a good representation $\mathbf{h}$ of a corrupted or partially destroyed representation of an input vector $\mathbf{x}$ is informative of $\mathbf{x}$ and invariant to a perturbation $\mathbf{x}^{(\text{corrupted})}$ of $\mathbf{x}$ due to noise. The problem considered in this paper is different since input features are extracted from already noisy real-life spoken documents. Such a noise is unpredictable and a noise model cannot be identified as the corruption of a clean input.

3.4. Proposed Deep Stacked Autoencoders (DSAE)

It has been argued in [24, 9] that DNNs may encode input data at progressively deeper levels of abstraction in successive hidden layers of stacked autoencoders. In a DNN of this type with k hidden layers, the latent features at the i-th intermediate hidden layer, for an input vector $\mathbf{x}$, are computed as the elements of a vector $\mathbf{h}^{(i)}$ as follows: $\mathbf{h}^{(i)} = \sigma(\mathbf{W}^{(i)}\mathbf{h}^{(i-1)} + \mathbf{b}^{(i)})\ \forall i \in \{1, \ldots, k\}$ and $\mathbf{h}^{(0)} = \mathbf{x}$. Therefore, each layer is pre-trained as a shallow autoencoder for a fixed number of iterations. The learnt hidden layer vector $\mathbf{h}^{(i)}$ is stored and used to learn the next layer $\mathbf{h}^{(i+1)}$. Greedy pre-training is progressively performed in this way starting with $\mathbf{h}^{(i+1)}$. After pre-training the last layer, a fine-tuning is performed on the entire stack of hidden layers to obtain a generative model providing different lev-

els of abstractions for the input vector $\mathbf{x}$.

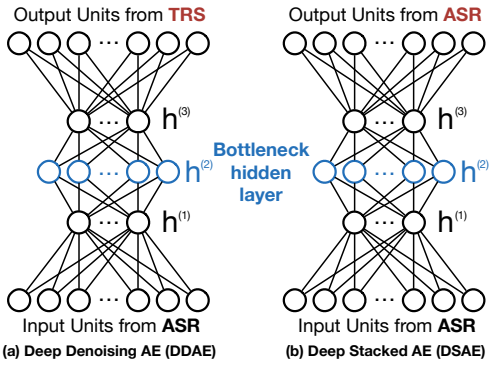

Figure 3: Illustration of the proposed bottleneck features (b) and a deep denoising autoencoder (DDAE) (a).

4. Experimental Protocol

The effectiveness of the proposed robust bottleneck features based on a deep stacked autoencoder is evaluated in the application framework of the DECODA corpus [25, 26, 27]. A classification approach based on a Multilayer Perceptron is performed to find out the main theme of a given real-life dialogue.

4.1. DECODA Corpus and input features

The corpus is a set of human-human telephone conversations from the customer care service of the RATP Paris transportation system from the DECODA project [25], used to perform experiments on the conversation theme identification. It is composed of 1,242 telephone conversations, corresponding to about 74 hours of signal, split as described in [25]. Each Conversation has been manually transcribed and labeled with one theme (on 8 possible themes) corresponding to the principal conversation concern. The train set is used to compose the subset of discriminative wordswith the TF-IDF-Gini method. For each theme, a set of 100 theme specific words is identified to form a vocabulary of 707 words. All the selected words are then merged without repetition and a same word may appear in more than one theme vocabulary selection.

4.2. Automatic Speech Recognition (ASR) System

The LIA-Speeral ASR system [28] with 230,000 Gaussians in the triphone acoustic models has been used for the experiments. The vocabulary contains 5,782 words. A 3-gram language model (LM) was obtained by adapting a basic LM with the transcriptions of the DECODA train set. The ASR system word error rate (WER) is 33.8% on the train, 45.2% on the development, and 49.5% on the test set. These high WER are mainly due to speech disfluencies and to adverse acoustic environments for some dialogues (calls from train stations, noisy/crowded streets with mobile phones...). A close WER has been reported on a similar corpus type (call-center conversations) [29].

4.3. Autoencoders Setup

Two autoencoders are trained as explained in Section 3.2. The input of the former, called AE_{ASR}, is a feature vector $\mathbf{x}^{(\text{ASR})}$ of noisy documents. The input of the latter, called AE_{TRS}, is a feature vector $\mathbf{x}^{(\text{TRS})}$ of clean documents.

Both autoencoders have one 50 nodes hidden layer $\mathbf{h}$. A deep stacked autoencoder (DSAE) (see Figure 3-(b)) is trained to extract features from the bottleneck layer $\mathbf{h}^{(2)}$. It uses noisy term-frequency vectors $\mathbf{x}^{(\text{ASR})}$ as input. The $\mathbf{h}^{(1)}$ and $\mathbf{h}^{(3)}$ hidden layers have both 50 neurons, the bottleneck $\mathbf{h}^{(2)}$ has 300 neurons and the size of the reconstructed output $\tilde{\mathbf{x}}^{(\text{ASR})}$ is equal to the size of the clean term-frequency vectors.
For comparison, a denoising autoencoder (DAE) is trained and evaluated. Its input is from ASR and output from TRS with a hidden layer of size 50 (see Figure 2) without the additional artificial noise (arrow between $\mathbf{x}^{(\text{corrupted})} = \mathbf{x}^{(\text{ASR})}$ and $\mathbf{x} = \mathbf{x}^{(\text{TRS})}$) and a deep denoising autoencoder (DDAE) with 3 hidden layers of the same size than the DSAE.
The robustness to noise of these different architectures is evaluated with the accuracy metric on DECODA theme identification task. Themes are hypothesized by a multi-layer perceptron (MLP) with one hidden layer with 256 nodes and an output layer with 8 neurons each corresponding to a theme. The Keras library [30], that uses Theano [31] for fast tensor manipulation and CUDA-based GPU acceleration, has been used to implement autoencoders, trained on the Nvidia GeForce GTX TITAN X GPU card. The MLP learning process lasts 8.33 minutes. The processing times of the architectures are as follows: 10 minutes for AE, 25 minutes for DAE, and 25 minutes for DSAE.

5. Experiments and Results

Experimental results on theme identification with shallow autoencoders (AE) reported in Section 5.1 show that a AE with input and output from ASR outperforms an hybrid denoising autoencoder. In Section 5.2, results obtained with deep autoencoders are compared with those obtained with the proposed deep stacked autoencoder.

5.1. Homogeneous and hybrid autoencoders

Table 1 presents theme classification accuracies with features from autoencoders obtained with two different transcription conditions (manual TRS or automatic ASR). For comparison, accuracies are reported with different MLP classification inputs: input $\mathbf{x}$, hidden $\mathbf{h}$ and output $\tilde{\mathbf{x}}$. It is worth emphasizing first that the best accuracies observed are obtained with hidden vectors for homogeneous conditions (ASR $\rightarrow$ ASR and TRS $\rightarrow$ TRS) with a gain of 3.9 and 0.7 points for ASR and TRS respectively. Accuracies obtained with the hybrid denoising autoencoder (DAE) from ASR $\rightarrow$ TRS decrease when the vector representation becomes more and more abstract ($\text{Acc.}(\tilde{\mathbf{x}}) < \text{Acc.}(\mathbf{h}) < \text{Acc.}(\mathbf{x})$). As expected, we can note that errors contained in the automatic transcriptions of noisy documents lower the accuracy, from 84.1% to 81% for the vector $\mathbf{h}$ for example. The accuracy of 84.1% obtained with the AE_{TRS} is the best classification result obtained with manual transcriptions. It represents the upper bound of the classification accuracy that can be obtained by denoising the ASR transcriptions (target of the approach proposed in this paper). The best results with ASR generated word hypotheses are obtained by feeding an MLP classifier with the latent representation h obtained with the DAE architecture shown in Figure 2. This suggests considering DDAEs with progressive denoising hidden layers to feed the highest classification layer with parameters trained with clean input as shown in Figure 3a. These DDAEs are compared with various types of bottleneck features in a deep stacked autoencoder DSAE as shown in Figure 3b.

Table 1: Theme classification accuracies (%) obtained at the output of an MLP classifier with input features x ,and h as shown in Figure 2 .

Method	Input data	Output data	Accuracy on test set		
			input **x**	hidden **h**	output $\tilde{\mathbf{x}}$
AE_{ASR}	ASR	ASR	77.1	**81**	79
AE_{TRS}	TRS	TRS	83.4	84.1	83.7
DAE	ASR	TRS	77.1	74.3	70.3

5.2. Deep denoising autoencoder (DDAE) *vs.* proposed deep stacked autoencoder (DSAE)

Table 2 compares the classification accuracies of features from the proposed deep stacked autoencoder (DSAE) and a deep denoising autoencoder (DDAE). The "Real" accuracies observed for the Test dataset are obtained depending on the Best accuracy reached by the Dev. set depending on the number of iterations during the learning process of the MLP classifier. This MLP uses input features h^n as well as $\tilde{\mathbf{x}}$ extracted by the architectures in Figure 3 with TRS output (Figure 3a) and ASR output (Figure 3b). The best results with ASR generated word hypotheses are obtained by feeding an MLP classifier with the latent representation that corresponds to the deepest level of abstraction before input reconstruction. Best results are obtained with DSAE that achieves a noteworthy accuracy of 82%, this result being quite close to the one reached by the AE with clean input and output (AE_{TRS}). A higher number of hidden layers reduces the classification performance, dropping in the worst case to 69.4% with DDAE's $\mathbf{h}^{(3)}$. The proposed DSAE obtains good accuracies with a gain of 9.5 points compared to DDAE. This is mainly due to the fact that DSAE is trained only with the ASR input/output, the mismatch reconstruction error with the clean desired output TRS not being back-propagated through its hidden layers. The proposed DSAE learns more robust features to noise by reaching a higher level of abstraction.

Table 2: Theme classification accuracies (%) obtained from an MLP with input features h^n and $\tilde{\mathbf{x}}$ extracted by the architectures in Figure 3 with TRS output (Figure 3a) and ASR output (Figure 3b).

Autoencoder employed	Layer vector	Best Dev Accuracy	Real Test Accuracy	Best Test Accuracy
Deep	$\mathbf{h}^{(1)}$	78	72.5	72.7
Denoising	$\mathbf{h}^{(2)}$	77.1	70.0	70.6
Autoencoder	$\mathbf{h}^{(3)}$	80.5	69.4	70.0
(**DDAE**)	$\tilde{\mathbf{x}}$	76.5	69.7	70.9
Deep	$\mathbf{h}^{(1)}$	87.0	81.7	82.8
Stacked	$\mathbf{h}^{(2)}$	88.0	**82.0**	83.0
Autoencoder	$\mathbf{h}^{(3)}$	87.4	80.1	81.9
(**DSAE**)	$\mathbf{h}^{(4)}$	87.0	81.0	83.1

6. Discussion

Table 3 compares the best accuracies of the different architectures and features proposed in this paper. Firstly, the AE_{ASR} and DSAE methods are among the more robust evaluated approaches. This can be explained by the fact that these neural networks are both able to remove an important portion of the noise contained in the documents. The classical AE with ASR for both input and output outperforms the DAE with heterogeneous conditions (input from ASR and output from TRS). This is mainly due to the fact that during the learning process, the quadratic error between the desired clean output and the ground truth output from TRS is back-propagated through all hidden layers. Indeed, the fist hidden layer $\mathbf{h}^{(1)}$ is an abstract representation of the ASR input vector. Thus, the error learned in the "clean" hidden space $\mathbf{h}^{(3)}$ and back-propagated through the bottleneck $\mathbf{h}^{(2)}$ layer inset a residual noise in the "clean" hidden spaces $\mathbf{h}^{(1)}$ and $\mathbf{h}^{(2)}$. However, both methods can not achieve the performance obtained with the original clean corpus.

Table 3: Best theme classification accuracy (%) observed for each set of features from ASR.

Method employed	Feature vector	Test Accuracy
DDAE	$\mathbf{h}^{(1)}$	69.4
DAE	**h**	74.3
Term-frequency	–	77.1
AE_{ASR}	**h**	81
Proposed DSAE	$\mathbf{h}^{(2)}$	**82.0**

The accuracy of the DSAE is of 82.0%, only 2.1 points under the accuracy obtained with clean documents (TRS) as presented in Table 1. This shows that a very small percentage of reconstructed feature vectors affect the classification performance of the target manual transcription features. Finally, the relatively low results (see Table 3) of DDAE, DAE methods show that trying to remove both noises at the same time is a bad idea. The conjugate noise in ASR documents is too complex to be directly removed. In the proposed DSAE, the first and last layers capitalize on the capacity of each hidden layer to remove residual noise by abstracting input representations in robust hidden spaces. Then, the mapping layer focuses on the more complex noise. This process lets this deep stacked AE produce a cleaner latent representation in the bottleneck hidden layer ($\mathbf{h}^{(2)}$). The 82% accuracy reported in Table 3 with bottleneck features compares favorably with the 81.4% reported in (Morchid 2014) for the best set of latent Dirichlet allocation (LDA) features selected with the development set. While specific tuning is not required for bottleneck features, the number of LDA hidden topics and the hyper-parameter values substantially affect accuracy with variations that are difficult to control.

7. Conclusion

This paper proposed an original document representation based on bottleneck features from a deep stacked autoencoder (DSAE) to address the difficult task of theme identification of automatically transcribed documents. The initial assumption to learn a deep autoencoder in homogeneous conditions (input/output with ASR vectors) confirmed by these experiments, allows us to better map documents into a latent reduced space, with a gain of more than 1 point compared to a shallow autoencoder with input/output from ASR. This is due to the small size of the dataset and to the only first hidden layer of the autoencoder that is enough to represent the relevant information. Moreover, the paper demonstrates that straightforward autoencoders leaned in homogeneous conditions outperforms accuracies obtained with denoising autoencoders learned with mismatched data (ASR and TRS as input/output). A future work to continue this preliminary study will be to take into account documents structure by replacing feed-forward layer in the DSAE with recurrent layer to use the various properties of recurrent neural networks such as Long-Short Term Memory (LSTM) autoencoders [32] or Gated Recurrent units [33].

8. References

[1] J. Eisenstein and R. Barzilay, "Bayesian unsupervised topic segmentation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. ACL, 2008, pp. 334–343.

[2] K. Lagus and J. Kuusisto, "Topic identification in natural language dialogues using neural networks," in *Proceedings of the Third SIGdial Workshop on Discourse and Dialogue*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, July 2002, pp. 95–102. [Online]. Available: http://www.aclweb.org/anthology/W02-1014

[3] G. Tur and R. De Mori, *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons, 2011.

[4] T. Hazen, "Topic identification," *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, pp. 319–356, 2011.

[5] I. Melamed and M. Gilbert, "Speech analytics," *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, pp. 397–416, 2011.

[6] M. Purver, "Topic segmentation," *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, pp. 291–317, 2011.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[8] D. Yu, L. Deng, and S. Wang, "Learning in the deep-structured conditional random fields," in *Proc. NIPS Workshop*, 2009, pp. 1–8.

[9] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[10] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[11] D. Yu, S. Wang, Z. Karam, and L. Deng, "Language recognition using deep-structured conditional random fields," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 5030–5033.

[12] A. Mohamed, G. Dahl, and G. Hinton, "Deep belief networks for phone recognition,[in:] nips workshop on deep learning for speech recognition and related applications," 2009.

[13] A.-r. Mohamed, D. Yu, and L. Deng, "Investigation of full-sequence training of deep belief networks for speech recognition." in *INTERSPEECH*, 2010, pp. 2846–2849.

[14] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *The Journal of Machine Learning Research*, vol. 11, pp. 625–660, 2010.

[15] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[16] J. Xu, L. Xiang, Q. Liu, H. Gilmore, J. Wu, J. Tang, and A. Madabhushi, "Stacked sparse autoencoder (ssae) for nuclei detection on breast cancer histopathology images," 2015.

[17] F. Camacho, R. Torres, and R. Ramos-Pollán, "Feature learning using stacked autoencoders to predict the activity of antimicrobial peptides," in *Computational Methods in Systems Biology*. Springer, 2015, pp. 121–132.

[18] J. Maria, J. Amaro, G. Falcao, and L. A. Alexandre, "Stacked autoencoders using low-power accelerated architectures for object recognition in autonomous systems," *Neural Processing Letters*, pp. 1–14, 2015.

[19] X. Zhou, J. Guo, and S. Wang, "Motion recognition by using a stacked autoencoder-based deep learning algorithm with smart phones," in *Wireless Algorithms, Systems, and Applications*. Springer, 2015, pp. 778–787.

[20] A. M. Sarroff and M. Casey, "Musical audio synthesis using autoencoding neural nets," 2014.

[21] L. Chao, J. Tao, M. Yang, and Y. Li, "Improving generation performance of speech emotion recognition by denoising autoencoders," in *Chinese Spoken Language Processing (ISCSLP), 2014 9th International Symposium on*. IEEE, 2014, pp. 341–344.

[22] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[23] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *The Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[24] Y. Bengio, Y. LeCun *et al.*, "Scaling learning algorithms towards ai," *Large-scale kernel machines*, vol. 34, no. 5, 2007.

[25] F. Bechet, B. Maza, N. Bigouroux, T. Bazillon, M. El-Beze, R. De Mori, and E. Arbillot, "Decoda: a call-centre human-human spoken conversation corpus." LREC'12, 2012.

[26] M. Morchid, M. Bouallegue, R. Dufour, G. Linares, D. Matrouf, and R. De Mori, "An i-vector based approach to compact multi-granularity topic spaces representation of textual documents." in *EMNLP*, 2014, pp. 443–454.

[27] M. Morchid, R. Dufour, and G. Linarès, "Topic-space based setup of a neural network for theme identification of highly imperfect transcriptions," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2015, Scottsdale, AZ, USA, December 13-17, 2015*, 2015, pp. 346–352. [Online]. Available: http://dx.doi.org/10.1109/ASRU.2015.7404815

[28] G. Linares, P. Nocéra, D. Massonie, and D. Matrouf, "The lia speech recognition system: from 10xrt to 1xrt," in *Text, Speech and Dialogue*. Springer, 2007, pp. 302–308.

[29] M. Garnier-Rizet, G. Adda, F. Cailliau, J. Gauvain, S. Guillemin-Lanne, L. Lamel, S. Vanni, and C. Waast-Richard, "Callsurf-automatic transcription, indexing and structuration of call center conversational speech for knowledge extraction and query by content," in *Proceedings of LREC*, 2008.

[30] F. Chollet, "Keras: Theano-based deep learning library," *Code: https://github. com/fchollet. Documentation: http://keras. io*, 2015.

[31] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, Jun. 2010, oral Presentation.

[32] K. Cho, B. v. M. C. Gulcehre, D. Bahdanau, F. B. H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation."

[33] A. Droniou and O. Sigaud, "Gated autoencoders with tied input weights," in *International Conference on Machine Learning*, 2013, p. x.