Journal of Biomedical Informatics 84 (2018) 103–113

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Patient representation learning and interpretable evaluation using clinical notes

Madhumita Sushil[a,b,*], Simon Šuster[b], Kim Luyckx[a], Walter Daelemans[b]

[a] Antwerp University Hospital, ICT Department, Wilrijkstraat 10, Edegem 2650, Belgium
[b] Computational Linguistics and Psycholinguistics (CLiPS) Research Center, University of Antwerp, Prinsstraat 13, Antwerp 2000, Belgium

ARTICLE INFO

Keywords:
Representation learning
Patient representations
Model interpretability
Natural language processing
Unsupervised learning

ABSTRACT

We have three contributions in this work: 1. We explore the utility of a stacked denoising autoencoder and a paragraph vector model to learn task-independent dense patient representations directly from clinical notes. To analyze if these representations are transferable across tasks, we evaluate them in multiple supervised setups to predict patient mortality, primary diagnostic and procedural category, and gender. We compare their performance with sparse representations obtained from a bag-of-words model. We observe that the learned generalized representations significantly outperform the sparse representations when we have few positive instances to learn from, and there is an absence of strong lexical features. 2. We compare the model performance of the feature set constructed from a bag of words to that obtained from medical concepts. In the latter case, concepts represent problems, treatments, and tests. We find that concept identification does not improve the classification performance. 3. We propose novel techniques to facilitate model interpretability. To understand and interpret the representations, we explore the best encoded features within the patient representations obtained from the autoencoder model. Further, we calculate feature sensitivity across two networks to identify the most significant input features for different classification tasks when we use these pretrained representations as the supervised input. We successfully extract the most influential features for the pipeline using this technique.

1. Introduction

Representation learning refers to learning features of data that can be used by machine learning algorithms for different tasks. Sparse representations, such as a bag of words from textual documents, treat every dimension independently. For example, in one-hot sparse representations, the terms ‘pain’ and ‘ache’ correspond to separate dimensions despite being synonyms of each other. Several techniques exist to model such dependence and reduce sparsity. The generalized or distributed representations learned using these techniques are referred to as low dimensional, or dense data representations. Unsupervised techniques for representation learning have become popular due to their ability to transfer the knowledge from large unlabeled corpora to the tasks with smaller labeled datasets, which can help circumvent the problem of overfitting [1].

Representation learning techniques have been used extensively within and outside the clinical domain to learn the semantics of words, phrases, and documents [2,3]. We apply such techniques to create a patient semantic space by learning dense vector representations at the patient level. In a patient semantic space, “similar” patients should have similar vectors. Patient similarity metrics are widely used in several applications to assist clinical staff. Some examples are finding similar patients for rare diseases [4], identification of patient cohorts for disease subgroups [5], providing personalized treatments [6,7], and predictive modeling tasks such as patient prognosis [8,9] and risk factor identification [10]. The notion of patient similarity is defined differently for different use cases. When it is defined as an ontology-guided distance between specific structured properties of patients such as diseases and treatments, it represents patient relationships corresponding to those properties. For example, if patient similarity is calculated as a hierarchical distance between the primary diagnostic codes of patients in the UMLS®metathesaurus [11], the value represents a diagnostic similarity. When it is defined as an intersection between the sets of blood tests performed on patients, patient similarity maps to blood test similarity. If patient similarity value is 1 for the patients of the same gender and 0 otherwise, groups of similar patients are gender-specific patient cohorts. However, when we calculate similarity between distributed patient representations, the different properties that influence the similarity value are unknown. Within the learned patient representations, we aim to capture similarity on multiple dimensions,

[*] Corresponding author at: Computational Linguistics and Psycholinguistics (CLiPS) Research Center, University of Antwerp, Prinsstraat 13, Antwerp 2000, Belgium.
E-mail address: madhumita.sushil@outlook.com (M. Sushil).

https://doi.org/10.1016/j.jbi.2018.06.016
Received 16 February 2018; Received in revised form 7 June 2018; Accepted 28 June 2018
Available online 03 July 2018

such as complaints, diagnoses, procedures performed, etc., which would encapsulate a holistic view of the patients.

In this work, we create unsupervised dense patient representations from clinical notes in the freely available MIMIC-III database [12]. We aim to learn patient representations that can later be used to identify sets of similar patients based on representation similarity. We focus on different techniques to learn patient representations using only textual data. We explore the usage of two neural representation learning architectures—a stacked denoising autoencoder [13], and a paragraph vector architecture [14]—for unsupervised learning. We then transfer the representations learned from the complete patient space to different supervised tasks, with an aim to generalize better on the tasks for which we have limited labeled data.

Dense representations can capture semantics, but at a loss of interpretability. Yet, it is critical to understand model behavior when statistical outputs influence clinical decisions [15]. We take a step towards bridging this gap by proposing different techniques to interpret the information encoded in the patient vectors, and to extract the features that most influence the classification output when these representations are used as the input.

2. Related work

Dense representations of words [16–19] and documents [14,20] have become popular because they are learned using unsupervised techniques, they capture the semantics in the content, and they generalize well across multiple tasks and domains. An *autoencoder* learns the data distribution and the corresponding dense representations in the process of first encoding data into an intermediate form and then decoding it. Miotto et al. [21] first proposed the use of a stacked denoising autoencoder to learn patient representations. They have shown promising results when patient vectors are first learned by a stacked denoising autoencoder from structured data combined with 300 topics from unstructured data, and are then used with Random Forests classifiers to identify future disease categories of patients. Following their work, Dubois et al. [22] have proposed two techniques to obtain patient representations from clinical notes. The first technique is unsupervised and performs an aggregation of concept embeddings into note and patient level representations, known as ‘embed-and-aggregate’. The second technique uses a recurrent neural network (RNN) with a bag-of-concepts representation of patient notes as time steps. The RNN is trained to predict disease categories of patients. The representations learned in this supervised setup are then transferred to other tasks. Apart from these works, Suresh et al. [23] have performed a preliminary exploration of the use of sequence-to-sequence autoencoders to induce patient phenotypes using structured time-series data. They have compared different autoencoder architectures based on their reconstruction error when they are trained to encode patient phenotypes. An application of these phenotypes to different clinical prediction tasks has been reserved for future work. In the same vein as these previous works, we investigate the applicability of a stacked denoising autoencoder to learn patient representations *directly from unstructured data*, and analyze the tasks that these representations can be successfully applied to.

One of the evaluation tasks for us is *patient mortality prediction*. Johnson et al. [24] provide a good overview of the previous approaches for mortality prediction on the MIMIC datasets with an aim of replicating the experiments. Following the work by Ghassemi et al. [25], Grnarova et al. [26] have shown significant improvements for mortality prediction tasks on using a two-level convolutional neural network (CNN) architecture, as compared to the use of topic models and doc2vec representations as inputs to linear support vector machines (SVMs). Besides these works, Jo et al. [27] have recently used long short term memory networks (LSTMs) and topic modeling for mortality prediction. They treat topics for patient notes as time steps for LSTMs. These topics are learned jointly using an encoder network. They have shown performance gains when the topics are jointly learned, compared to those pretrained using LDA [28].

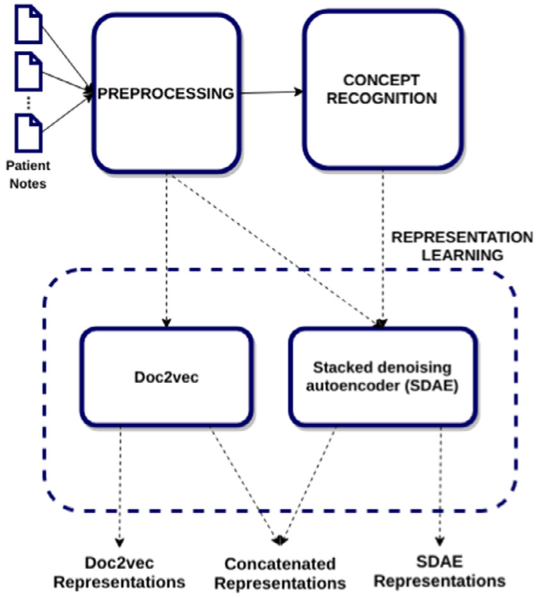

Fig. 1. An overview of the patient representation pipeline. The dashed lines indicate one of several operations, and are not performed in parallel.

3. Methods

3.1. Learning patient representations

In this section, we describe a stacked denoising autoencoder and a paragraph vector architecture doc2vec, in the context of learning task-independent dense patient representations in an unsupervised manner. The corresponding methodology for learning these dense representations is illustrated in Fig. 1.

3.1.1. Stacked denoising autoencoder

Given the previous success of autoencoders for representation learning using structured data with or without topic models learned from unstructured data, we explore the use of a stacked denoising autoencoder (SDAE) [13] to learn task-independent patient representations from raw clinical text, forgoing the use of intermediate techniques like topic modeling. Although the premise of learning patient representations using an SDAE is not novel in itself, our contribution lies in analyzing if such a model is also successful when used only with clinical notes, and if the learned representations can be successfully applied for a range of tasks that are different from patient prognosis. This analysis gives us insight into successful and transferable patient representation architectures for unstructured data.

During the **pretraining** phase, every layer of an SDAE is sequentially trained as an independent denoising autoencoder. An autoencoder learns to first encode the input data I into an intermediate representation R, and then decode R into I. Denoising refers to the process of first adding noise to corrupt the input I into $\widetilde{I}$, and then training an autoencoder to reconstruct I using $\widetilde{I}$ as the input. We use the dropout noise [29], where a random proportion of the input nodes are set to 0. In the process of denoising, the model also learns the data distribution. In an SDAE, the intermediate representations obtained from the autoencoder at layer $n-1$ are used as the uncorrupted input to the autoencoder at layer n, for all the layers in the SDAE. To pretrain patient representations using an SDAE, high-dimensional (sparse) patient data are used as the input to the autoencoder at the first layer of the SDAE. The intermediate representations obtained from the autoencoder at the final layer are treated as the low-dimensional (dense) representations $R(p)$ for a patient p. The number of layers is determined through a random search [30] based on the results for primary diagnostic category prediction using a perceptron.

Finetuning can be performed in multiple ways [1]. In one approach, all the encoder layers can be stacked on top of each other, and a logistic regression layer can be added on the top to finetune the entire pretrained network for an end task as a feedforward neural network. In such a setup, the input features in the finetuning phase are the same as the input features during the pretraining phase. In another approach, instead of the entire network, only the preliminary task-independent representations R can be finetuned for an end task. In this approach, R is used as the input to a separate classifier. In our experiments, we train separate classifiers for different tasks using R as the input features.

We use the sigmoid activation function for the encoding layers, and the linear activation function to decode real values. During the pretraining phase, we train each layer of the SDAE to minimize the mean squared reconstruction error using the RMSProp optimizer [31]. During the finetuning phase, we train the classifiers to minimize the categorical cross-entropy error using the same optimizer. We determine the number of layers, the dimensionality, and the dropout proportion also using a randomized hyperparameter search. These values are dependent on the feature sets and the finetuning process, and can be found in Table A.1 in the Appendix.

3.1.2. *Paragraph vector*

Doc2vec, or 'Paragraph Vector' [14], learns dense fixed-length representations of variable length texts such as paragraphs and documents. It supports two algorithms—a distributed bag-of-words (DBOW) algorithm, and a distributed memory (DM) algorithm. For both the algorithms, word representations are shared among all the occurrences of a word across all the paragraphs, and paragraph vectors are shared among all the contexts that occur in a given paragraph. In the DBOW algorithm, word and paragraph vectors are jointly trained when the paragraph vectors are used to predict the context words for all the contexts in the paragraph. In the DM algorithm, these vectors are jointly trained by predicting the next word from a concatenation of the paragraph vectors and the vectors of the context words. During the inference phase of both the algorithms, word vectors are fixed, and paragraph vectors are trained until convergence.

We use the DBOW algorithm for 5 iterations, with a window size of 3, a minimum frequency threshold of 10, and 5 negative samples per positive sample to train 300-dimensional patient vectors. We determined these settings also using randomized hyperparameter search.

3.2. *Feature extraction*

When statistical models are deployed for clinical decision support, it is crucial to understand the features that influence the model output [15]. A ranked list of the most influential features can assist such understanding, while facilitating error analysis; it can also enable exploratory analysis when unexpected features are ranked high. However, neural networks are notorious for being black boxes due to their complex architectures. Given the impact of automated decisions, there has been a recent surge of interest to make neural architectures interpretable. Different techniques include visualization of weights and embeddings [32,33], representation erasure and feature occlusion [34,35], input perturbation [36], and visualization of attention weights in recurrent neural networks [37–40]. The technique of visualizing hidden weights and embeddings is a qualitative approach to interpretability. Furthermore, techniques like input feature erasure train a new model in absence of a given feature. When retrained, these models can learn to rely on a completely different set of features. Moreover, the attention mechanism is not applicable to feedforward neural networks. Within the scope of our work, we propose two techniques to bridge the existing gap in model interpretability when we train unsupervised dense representations, and when we use these representations to get classification decisions using feedforward neural networks. To the best of our knowledge, we are the first to propose these techniques to make dense representations interpretable.

3.2.1. *Average feature reconstruction error: pretraining phase*

We calculate the **squared reconstruction error** of all the input features in the first layer of the pretrained autoencoder, averaged across all the training instances. The value of the reconstruction error of the individual features gives us an estimate of the features that are encoded the best and the worst in the patient vectors learned through the SDAE. This knowledge facilitates an analysis of model behavior to make the vectors more interpretable.

3.2.2. *Input significance calculation using sensitivity analysis: classification phase*

Sensitivity analysis, or gradient-based analysis, is often used to identify the most influential features of a trained model [41–43]. For a given model and a given instance, the sensitivity of an output node with respect to an input node refers to the observed variation in the output on varying the input. This is equivalent to the gradient of the output with respect to the input. The inputs that cause larger variations in the output are more significant for the model.

This analysis has so far been used to identify the most influential features for a single network, such as a single classifier. However, in our work, we are confronted with two neural networks. The first network learns the dense patient representations, and the second network uses these dense representations as the input for different classification tasks. We extend the work by Engelbrecht and Cloete [41] and propose a technique to compute the significance of the original (sparse) features on the final classification decisions. We use the chain rule across two networks to compute the sensitivity of the output node in the second network to the input of the first network. This allows us to identify the most influential features in the entire pipeline.

We demonstrate this technique for different classification tasks when the task-independent dense patient representations R are first induced by the SDAE from the original input z, and R is then used as the input to the classifiers. The significance of the ith input feature (ϕ_{z_i}) is defined as the maximum significance of the input feature i across all the K output units (o) of the classifier with respect to the N instances:

$$\phi_{z_i} = \max_{k=1\dots K} \{S_{o_k z_i}\} \text{ where} \tag{1}$$

$$S_{o_k z_i} = \sqrt{\frac{\sum_{j=1}^{N} [S^{(j)}_{o_k z_i}]^2}{N}}. \tag{2}$$

$S^{(j)}_{o_k z_i}$ is the sensitivity of the kth output unit of the classifier w.r.t the ith input feature of the SDAE for an instance j:

$$S^{(j)}_{oz,ki} = \frac{\partial o^{(j)}_k}{\partial z^{(j)}_i} = \frac{\partial o^{(j)}_k}{\partial R^{(j)}_i} * \frac{\partial R^{(j)}_i}{\partial z^{(j)}_i}. \tag{3}$$

In (2), we thus calculate the mean squared sensitivity across different N instances and take the root. The sensitivity for a particular instance (3) is obtained by first taking the derivative of an output node value w.r.t. a value in a patient representation; then taking the derivative of the patient representation value w.r.t. the original input value; and then multiplying them. This technique allows us to identify the most significant features in a trained model for an arbitrary number of instances and output classes. It is also transferable to the doc2vec representations, but we reserve this for future research.

4. Dataset construction and preprocessing

We retrieve a set of adult patients (⩾18 years age) with only one hospital admission, with at least one associated textual note (excluding discharge reports) from the MIMIC-III critical care database [12]. We restrict to the patients with a single admission to remove ambiguity

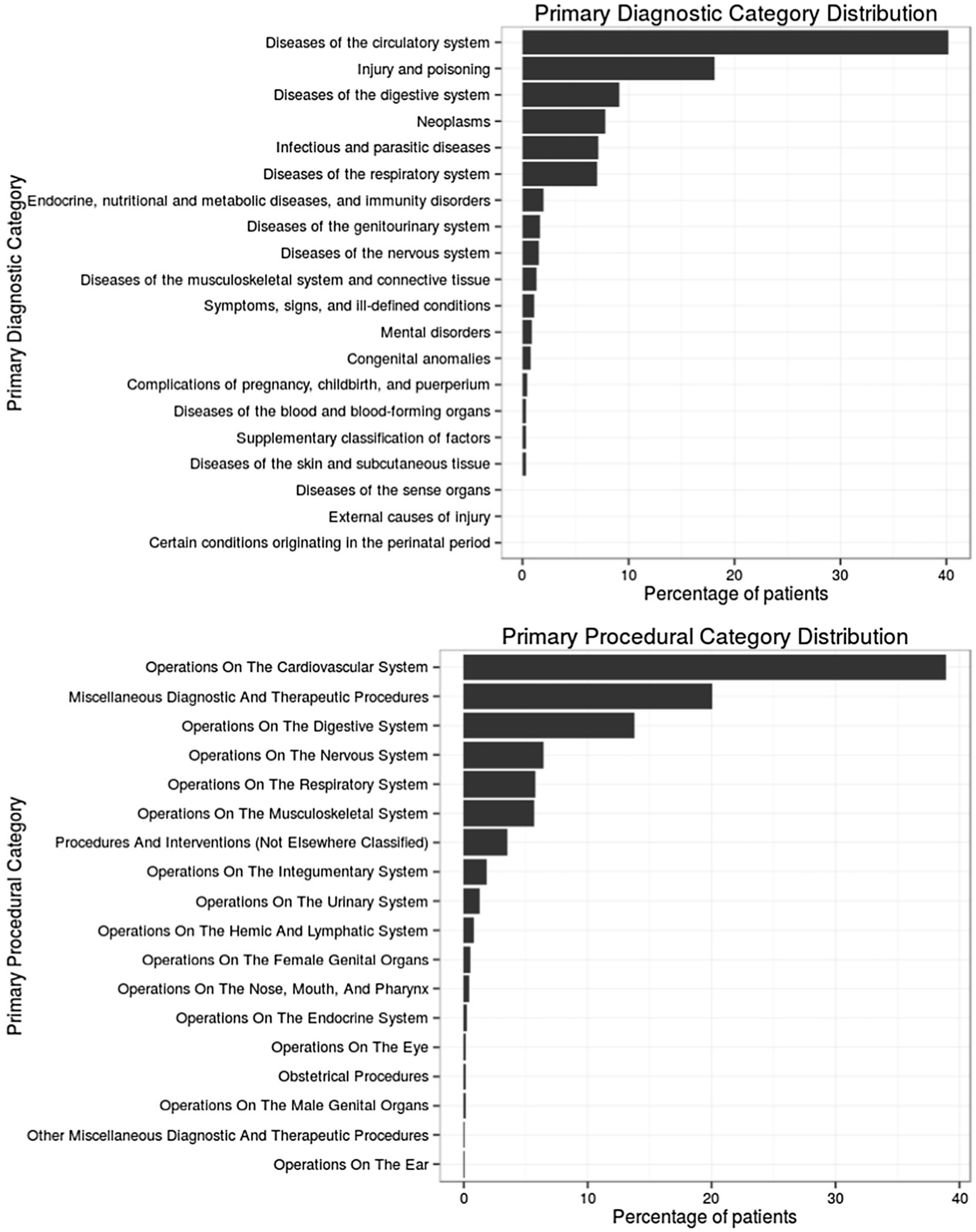

Fig. 2. Primary diagnostic and procedural category distribution in the data.

when the labels are dependent on discharge time. We exclude discharge reports from analyses to remove the direct indication of in-hospital death of a patient, which is one of the tasks that we are interested in. We obtain a range of 1–879 notes per patient, with average of 29.51 notes. This corresponds to 13–789,906 tokens per patient, with an average of 13,064 tokens. We split the dataset into 80-10-10% as training, validation, and test subsets, to get a set of 24,650 patients for training, and 3081 patients each for validation and testing. We represent patients with a concatenation of all the notes associated with them (excluding discharge reports). We tokenize the dataset using the Ucto tokenizer [44] and lowercase it.

To obtain patient representations using the SDAE and for the baseline experiments, we replace the numbers, and certain token-level time and measurement matches with placeholders. We remove the punctuations, and the terms with corpus frequency less than 5. We represent the out-of-vocabulary terms obtained after the preprocessing in the test set with a common token. We use two feature sets—a bag-of-words (BoW), and a bag-of-medical-concepts (BoCUI)—with their corresponding TF-IDF scores as feature values. We use the TF-IDF values to give high weights to frequent features for a patient relative to all the patients in the dataset. For the BoCUI, we use the CLAMP toolkit [45] to identify Concept Unique Identifiers (CUIs) in the UMLS®metathesaurus [11] corresponding to medical concept mentions of the types problems, treatments, and tests as defined in the i2b2 annotation guidelines [46], along with their assertion labels. Here, problems also include findings and symptoms. CUIs appended with 'present' and 'absent' assertion labels are the vocabulary terms for this feature set. A bag-of-medical-concepts is a common featurization technique used in clinical NLP research [21,47]. We use a bag representation instead of a sequence model because the final document length for different patients is highly variable, going up to very large document sizes. We obtain a vocabulary size of 71,001 for the BoW feature set, and 83,310 for the BoCUI feature set.

To train the doc2vec models, we remove the numbers and the tokens matching certain time and measurement regex patterns. We have determined these settings based on the initial results on the validation set. We obtain a vocabulary size of 48,950 for this model. We have not trained a doc2vec model using only the medical concepts because if we represent a document as a sequence of CUIs only, we remove the indicators of language semantics from the context window, which the doc2vec model relies on during the learning process. If we keep additional terms along with the concept identifiers to train a doc2vec model, the available information is not comparable to a BoCUI feature set.

5. Evaluation

5.1. Task description

We use the dense patient representations as input features to train feedforward neural network classifiers on multiple independent tasks. We evaluate the performance on a range of tasks to gain insight into the task independent nature of the representations, and the information encoded within the vectors. We disregard the instances that do not have a task label. We minimize the categorical cross-entropy error using the RMSProp optimizer, and determine the hyperparameters using randomized search, which can be found in Table A.2 in the Appendix.

1. **Patient mortality prediction:** Whether a patient dies within a given time frame. This prediction gives an estimate of the severity of a patient's condition to decide the amount of attention required.
   (a) **In-hospital mortality (In_hosp):** Patient death during the hospital stay—13.14% of the instances in the dataset.
   (b) **30 days mortality (30_days):** Patient death within 30 days of discharge—3.85% of the instances in the dataset.
   (c) **1 year mortality (1_year):** Patient death within 365 days of discharge—12.19% of the instances in the dataset. This includes the patients who died within 30 days of discharge.
2. **Primary diagnostic category prediction (Pri_diag_cat):** Correctly diagnosing patients is essential for deciding further course of action. We evaluate if the proposed technique can be used to predict the generic category of the most relevant diagnostic code for a patient, corresponding to the 20 categories in the first volume of the 9th revision of the International Classification of Diseases, Clinical Modification (ICD-9-CM) database [48]. A distribution of these categories in the dataset is given in Fig. 2.
3. **Primary procedural category prediction (Pri_proc_cat):** Predicting the generic category of the most relevant procedure performed on a patient, corresponding to the 18 categories present in the third volume of the ICD-9-CM database. A distribution of these categories in the dataset is given in Fig. 2. These procedural categories reflect different surgeries performed on patients. Prediction of the recommended procedure would assist the medical staff, while enabling optimal resource allocation for the same.
4. **Gender:** Gender of a patient—male (56.87% of the instances) or female (43.13% of the instances), as encoded in the dataset.

We evaluate the models using the area under the ROC curve (AUC-ROC) for patient death for the mortality tasks. The ROC curve gives us insight into the trade-off between the true positive rate and the false positive rate at different thresholds for different models. For the other tasks, we compute the weighted F-score to correct for class imbalance. We present the classification pipeline in Fig. 3.

5.2. Results and discussion

5.2.1. Supervised representation evaluation

In Table 1, we compare the classification performance when we use the dense patient representations obtained from the SDAE-BoW (the initial SDAE input is BoW), the SDAE-BoCUI (the initial SDAE input is BoCUI), and the doc2vec models as input features for different tasks, as opposed to using the BoW and the BoCUI sparse features. In Fig. 4, we show the ROC curves for the mortality prediction tasks. Further, we analyze the agreement between the SDAE-BoW and the doc2vec model outputs by calculating Cohen's κ score [49] between them on the validation set. We find that the agreement scores are not high, which may indicate that the models learn complimentary information. We then concatenate the two dense representations (model ensemble) to analyze model complementarity. We calculate the statistical significance between the 9 different feature sets for the 6 tasks using the two-tailed pairwise approximate randomization test [50] with a significance level of 0.05 before the Bonferroni correction for 54 hypotheses.[1]

Our main finding is that all the dense representation techniques significantly outperform the BoW baseline for 30 days mortality prediction. However, although we see a large numerical improvement over the BoW baseline on using the dense representations for 1 year mortality prediction (where the set of instances with the label 'death' is a superset of those for 30 days mortality), the differences are not statistically significant. The SDAE-BoCUI model is significantly better than the BoCUI model for both 30 days and 1 year mortality prediction tasks. We believe that the poor performance of the sparse models for 30 days mortality prediction may be due to the low number of positive instances. The generalization afforded by the dense representation techniques assists feature identification in such cases. The sparse BoW inputs perform better than the SDAE-BoW representations for all the other tasks, and better than the doc2vec representations for in-hospital mortality and primary procedural category prediction. One probable reason is that the best predictors for the other tasks are the direct lexical mentions in the notes, which makes the BoW model a very strong baseline. Examples of such features obtained using the χ^2 feature analysis are 'autopsy', 'expired', 'funeral', and 'unresponsive' for in-hospital mortality prediction, and 'himself', 'herself', 'ovarian', and 'testicular' for gender prediction. It is interesting to point out that the direct mentions of in-hospital death are present in the notes even though discharge reports have been excluded from analysis.

The agreement scores between the doc2vec and the SDAE-BoW models are not high for any task, which may indicate that the two models are complementary to each other. The results obtained from concatenation of the vectors learned by both models is not significantly different from the sparse representations for any task except 30 days mortality prediction, where the concatenation is better. This ensemble model significantly outperforms both individual models for primary procedural category prediction. For primary diagnostic category and gender prediction, the ensemble model is significantly better than the SDAE model, but not the doc2vec model. In these cases, there is no significant difference between the doc2vec and the BoW models. Hence, we observe that the concatenation helps in some cases and we recommend combining the two dense representations for unknown tasks. The doc2vec model uses a local context window in a log-linear classifier, whereas the SDAE model uses only the global context information and non-linear encoding layers. This may be one of the factors governing the differences between the two techniques.

Furthermore, we observe that the BoCUI sparse features perform significantly worse than the BoW sparse features for in-hospital

[1] These hypotheses are the comparisons of the doc2vec, the SDAE-BoW, and the ensemble dense representations respectively with the BoW model, the ensemble with the doc2vec model, the ensemble with the SDAE-BoW model, the BoCUI with the BoW models, the SDAE-BoW model with the SDAE-BoCUI model, and the BoCUI model with the SDAE-BoCUI model for the 6 tasks.

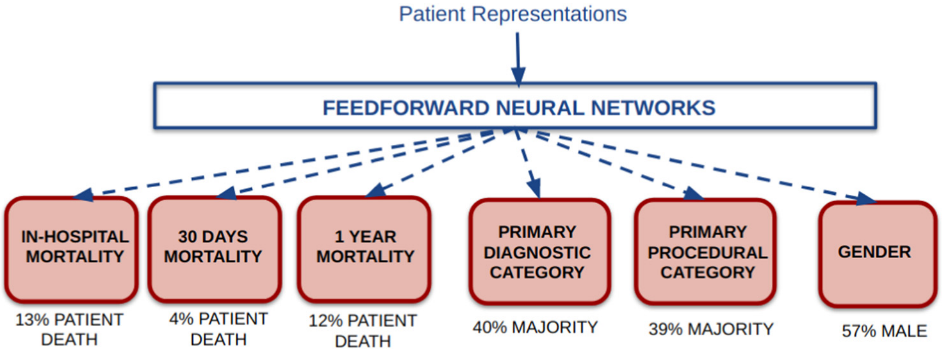

Fig. 3. Representation evaluation pipeline. The dashed lines indicate one of several operations, and are not performed in parallel.

Table 1
Classification results on different tasks using the BoW features, the SDAE representations computed from the BoW (SDAE-BoW), the doc2vec representations, the concatenated SDAE-BoW and doc2vec representations ([doc2vec, SDAE-BoW]) with Cohen's ϰ score (in italics), the BoCUI features, and the SDAE vectors computed from the BoCUI (SDAE-BoCUI). AUC-ROC values are reported for the mortality tasks, and weighted F-score for the others.

No.	Approach	Mortality			Pri_diag_cat	Pri_proc_cat	Gender
		In_hosp	30_days	1_year			
(1)	BoW	0.9457	0.5949	0.7942	0.7016	0.7366	0.9847
(2)	SDAE-BoW	0.9194	0.7965	0.7980	0.6500	0.6746	0.8775
(3)	doc2vec	0.9195	0.7680	0.8134	0.6807	0.6583	0.9770
(4)	[doc2vec, SDAE-BoW]	0.9383	0.8113	0.8302	0.6788	0.7030	0.9747
	(ϰ)	*0.5865*	*0.0000*	*0.1581*	*0.6438*	*0.5891*	*0.7200*
(5)	BoCUI	0.9088	0.5065	0.6993	0.7104	0.7265	0.7504
(6)	SDAE-BoCUI	0.9007	0.7832	0.8016	0.6647	0.6777	0.6245

mortality, 1 year mortality, and gender prediction. For the other tasks, there is no statistical difference between the performance of the BoW and the BoCUI features, although we see a large numerical drop of about 9% with the BoCUI model for 30 days mortality prediction. Moreover, the SDAE-BoW and SDAE-BoCUI representations are also not significantly different from each other for any of the tasks. These results suggest that there is no advantage of using a bag-of-concepts over a bag-of-words feature set, either as sparse inputs, or to learn dense representations. There are a few possible reasons behind the observed performance drop on using the BoCUI feature set. First, these features are restricted to the medical concepts of types 'problem', 'treatment', and 'test'. These concepts are important features for diagnostic and procedural category identification. However, when we remove the terms that do not belong to these types, we also remove some useful features for other tasks, e.g., pronouns for gender prediction, and terms like 'expired' and 'post-mortem' for in-hospital mortality prediction, which in turn affects the classification performance. Next, when we identify medical concepts mentions with their corresponding CUIs and assertion labels, we also propagate the errors along in the pipeline, while adding to the sparsity of the terms. These factors additionally contribute to a difference in the classification performance.

Our work on mortality prediction is related to Grnarova et al. [26]. The closest comparison between our results is the evaluation of the doc2vec representations. They have reported the AUC-ROC scores of 0.930, 0.831, and 0.824 for in-hospital mortality, 30 days mortality, and 1 year mortality prediction respectively, and have shown an improvement over the LDA baseline for the latter two. These scores are higher than what we have obtained with doc2vec. However, this may be due to different data subsets[2], different classifiers (feedforward neural networks vs. linear SVMs), or different training schemes. They have further reported significant improvement on all the tasks when using a CNN architecture. This setup is supervised for the mortality tasks, and it is unclear whether supervision plays a role in the observed improvement. Similarly, Jo et al. [27] have shown significant improvements for mortality prediction tasks on using their supervised LSTM architecture that jointly learns topic models as opposed to using LDA with linear SVMs. Again, the results are not directly comparable. They have predicted in-hospital, 30 days post-discharge, and 1 year post-discharge mortality at the end of every 12 h window during a patient stay. Instead, we predict these mortality values using all the notes (except discharge reports) until the end of the patient stay. They have not reported the AUC-ROC scores for patient mortality at the end of the patient stay.

Furthermore, Dubois et al. [22] have evaluated their embed-and-aggregate and RNN architectures for patient representation learning on multiple tasks. They have found that the RNN trained in a supervised manner for diagnostic code prediction outperforms the other architectures for predicting future diagnostic codes. However, when these representations are transferred to other tasks, this advantage is not visible. For mortality prediction (within the time period of the patient records) on large datasets, the bag-of-concepts and embed-and-aggregate methods performed equally well, and outperformed the RNN architectures. The RNN architecture performed poorly also for prediction of future patient admission, and had a comparable performance to embed-and-aggregate method for future ER visit prediction. One explanation for better RNN performance for future diagnostic code prediction is that the representations obtained from the RNN encode important information about patient diagnoses due to their supervised training on a similar task. This is not the case for the other tasks where there is no improvement.

5.2.2. *Feature analysis*

In Table 2, we present a list of features based on their mean squared reconstruction error when we pretrain the patient representations using the SDAE-BoW model. We observe that infrequent terms such as spelling errors are reconstructed very well, as opposed to the frequent

[2] We were unable to reconstruct exact data subsets and obtain comparable results because we did not have access to their data processing scripts and the complete pipeline.

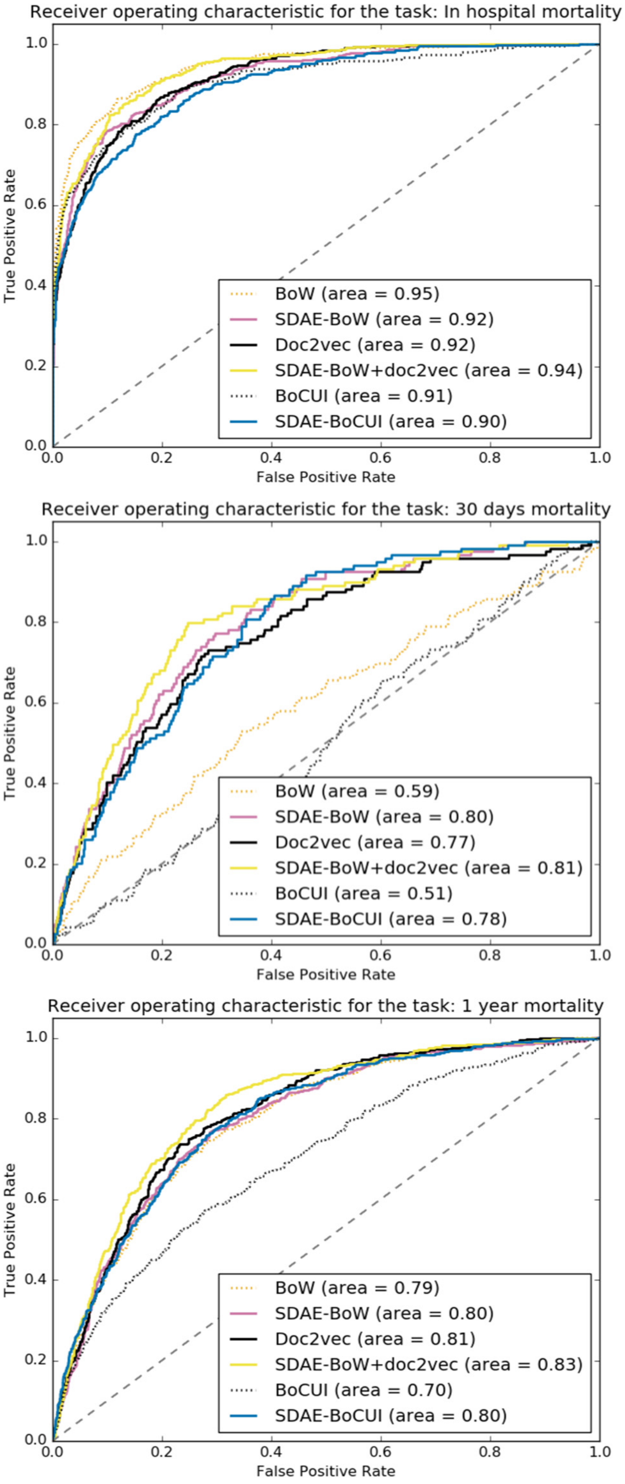

Fig. 4. Receiver operating characteristic (ROC) for patient mortality prediction tasks.

Table 2
The best and the worst feature reconstructions during unsupervised pretraining of SDAE-BoW.

Best reconstruction	Worst reconstruction
stumnz	picc
jajhnx	woman
a-fibril	osh
lsc.o	fall
potentiallly	man
yesh	stent
forcal	he
contbributing	wife
hyponatremia-on	repair
pre-exiusting	bleed

Table 3
Correlation between the mean squared reconstruction error of the first layer of the SDAE during the unsupervised pretraining phase and feature frequency. All the p-values are lower than 0.001.

Feature set	Spearman	Kendall-Tau
BoW	0.8738	0.7287
BoCUI	0.8836	0.7334

features in the dataset. To check for a correlation between the mean squared reconstruction error and the feature frequency, we calculate the Spearman's and the Kendall-tau rank-order correlation coefficients [51] between the two parameters, reported in Table 3. These techniques check for a correlation between the parameters irrespective of a linear relationship and use different algorithms to generate the ranked lists in case of a tie. Using both techniques, we obtain very high positive correlation coefficients. We believe that this behavior may be either due to the high entropy of the frequent terms, or because the model memorizes the infrequent terms. Jo et al. [27] also obtain misspellings and rare words as the top features when they use recurrent neural networks for patient mortality prediction in the MIMIC-III dataset.

In Table 4, we list the most significant features for the model output for one instance each in the test set, when the SDAE-BoW patient representations *R* are used as the classification input. In italics are the vocabulary terms that are not present in the notes for the patient, but are treated as the most influential features. We find that the classifiers give high importance to sensible frequent features for most of the tasks, although the SDAE reconstructs low frequent terms such as spelling errors better during the pretraining phase. Several features for in-hospital mortality point towards the overall patient condition and treatments for the patient. Terms like ‘brbpr’ (bright red blood per rectum) for primary diagnostic category prediction, and the top features for gender prediction indicate the true class. The absence of several features is used as an important clue to identify the right class. For example, most of the top ranking features for 30 days and 1 year mortality prediction are not present in the patient notes. Similarly, the absence of the terms related to the female gender implies the male class. Additionally, the absence of numbers (‘numeric_val’) in notes is the most useful feature for diagnostic and procedural category identification, which may have been used by the model to identify certain lab tests with numeric results that were not carried out.

Furthermore, many top features extracted for primary diagnostic category prediction are the terms corresponding to text segments like *“Sinus rhythm. Compared to the previous tracing of …”*, which is a common pattern in the notes for the patient. When evaluated without the context, many of these terms do not make sense. However, although we input a bag-of-words representation to the SDAE, co-occurrence of the terms is reflected in the extracted features. We further observe that there is a minimal overlap between the sets of important features for different tasks. This shows that the learned representations *R* are task-independent, and that the classifiers can identify task-specific important information when they are trained for a particular task.

To illustrate the applicability of the feature extraction technique to

Table 4
The most significant features in ranked order for the classifiers for one instance each when the SDAE-BoW representations are used as the input. The true classes are 'patient death' for the mortality tasks (a common instance for 30 days and 1 year mortality prediction), and 'diseases of the digestive system', 'operations on the digestive system', and 'male' respectively for a common patient for the other tasks.

In_hosp	30_days	1_year	Pri_diag_cat	Pri_proc_cat	Gender
vasopressin	leaflet	*magnevist*	*numeric_val*	*numeric_val*	*woman*
pressors	*structurally*	*signal*	previous	no	*female*
focused	*pacemaker*	*decisions*	rhythm	of	*she*
dnr	*sda*	*periventricular*	no	*enzymes*	man
dopamine	*periventricular*	*embolus*	*flexure*	*extubated*	he
acidosis	*excursion*	*underestimated*	*dementia*	rhythm	male
levophed	*non-coronary*	*calcified*	brbpr	and	*her*
pressor	*dosages*	*screws*	of	the	his
cvvhd	*microvascular*	*rib*	sinus	*vent*	wife
cvvh	left-sided	*shadowing*	for	*uncal*	*uterus*
emergency	*chronic*	*gadolinium*	to	*mso*	him
pneumatosis	*extubation*	*mri*	tracing	to	*urinal*

Table 5
Comparison of the best features for one instance of in-hospital patient death, where the BoW model makes the correct prediction and the SDAE-BoW model fails, and for one instance where both the models make the correct prediction.

BoW (correct)	SDAE-BoW (correct)	BoW (correct)	SDAE-BoW(correct)
expired	cad	expired	vasopressin
autopsy	cabg	autopsy	pressors
cmo	pre-op	morgue	focused
pre-bypass	preop	cmo	dnr
morgue	numeric_val	toradol	dopamine
diseasecoronary	no	diseasecoronary	acidosis
deline	bypass	deline	levophed
prebypass	sternotomy	prebypass	pressor
death	lat	pre-bypass	cvvhd
decannulation	ptx	asystolic	cvvh

understand relative model behavior, we compare the set of the most important features for a) one instance where the bag-of-words model predicts in-hospital death correctly, whereas the SDAE dense representations fail to make that prediction, and b) one instance where both the models make correct predictions. These features are presented in Table 5. We find that the BoW model identifies the direct indicators of patient death such as 'expired', 'autopsy', 'morgue', and 'death' as the top features along with certain features related to the procedures performed on the patient. Instead, the generalized SDAE-BoW model uses the features related to the holistic patient condition as the more important features. Examples are 'cad (Coronary Artery Disease)', 'cabg (Coronary Artery Bypass Graft surgery)', 'vasopressin', 'dopamine', 'dnr (do not resuscitate)', and 'cvvhd (Continuous Veno-Venous Hemofiltration Dialysis)'. This shows us that the models operate in very different feature spaces. The generalized models are good when we want a comprehensive view of the patient condition. However, the sparse BoW model may be better if we want to pick up the strong lexical features present for a task.

5.2.3. Visualization of unsupervised representations

In Fig. 5, we present 2D visualizations of the unsupervised representations learned by the SDAE and the doc2vec architectures. It is important to note that the SDAE-BoW and the doc2vec representations were learned in an unsupervised manner, and were not finetuned to represent a particular property of the data. Hence, they encode information that represent patient notes in a holistic manner, and span many different properties. We use t-SNE[3] [52] to generate the visualizations, after first reducing the representations to 50 dimensions[4] using Principal Component Analysis. In the figure, as an example, we color the representations according to the corresponding primary diagnostic category. For the purpose of clarity, we limit to the 5 most frequent diagnostic categories in the dataset. We observe that the patients with the same diagnostic category are frequently close together, forming clusters. This suggests that using the proposed techniques, "similar" patients result in similar representations.

6. Conclusions and future work

Our research provides insight into the suitability of learning patient representations only from clinical notes, for an arbitrary task, while understanding model performance. We have shown that the generalized dense patient representations significantly improve the classification performance for 30 days mortality prediction, a task where we are confronted with a very low proportion of positive instances. For the other tasks, this advantage is not visible. Moreover, we have shown that a combination of the stacked denoising autoencoder and the doc2vec representations improves over the individual models for some tasks, without any harm to the others tasks. We recommend combining these representations for unknown tasks. We have further shown that there is no advantage of using a bag-of-concepts feature set as opposed to a bag-of-words feature set as either sparse inputs or to learn dense representations. Expensive concept identification process is not required for these setups.

Furthermore, we have proposed novel techniques to interpret model performance to overcome the black-box nature of neural networks. During representation analysis, we have found that frequent terms are not encoded well during the pretraining phase of the stacked denoising autoencoder. However, when we use these pretrained vectors as the input, sensible frequent features are selected as the most significant features for the classification tasks. Some vocabulary items that are absent from patient notes are often deemed important, while at the same time, co-occurrence of the features present in the notes is also learned by the model. We have also shown that the unsupervised representations are task-independent and distinct features are extracted for different tasks when these representations are used as supervised inputs.

This work lays down the path for more applied research in the clinical domain. In future, we plan to compute patient similarity from the generalized patient representations to identify patient cohorts. We also plan to add structured information to analyze their comparative contribution to the learned representations for the different tasks. Furthermore, the techniques that we have proposed to understand the

[3] We experimented with different values of perplexity and the number of iterations for the t-SNE. After converging at 5000 iterations, the resulting visualizations were similar across most perplexity values, albeit often rotated. We chose a perplexity of 50 for the SDAE-BoW representations, and 30 for the doc2vec representations.

[4] Nearly 70% of the variation was explained by these 50 dimensions.

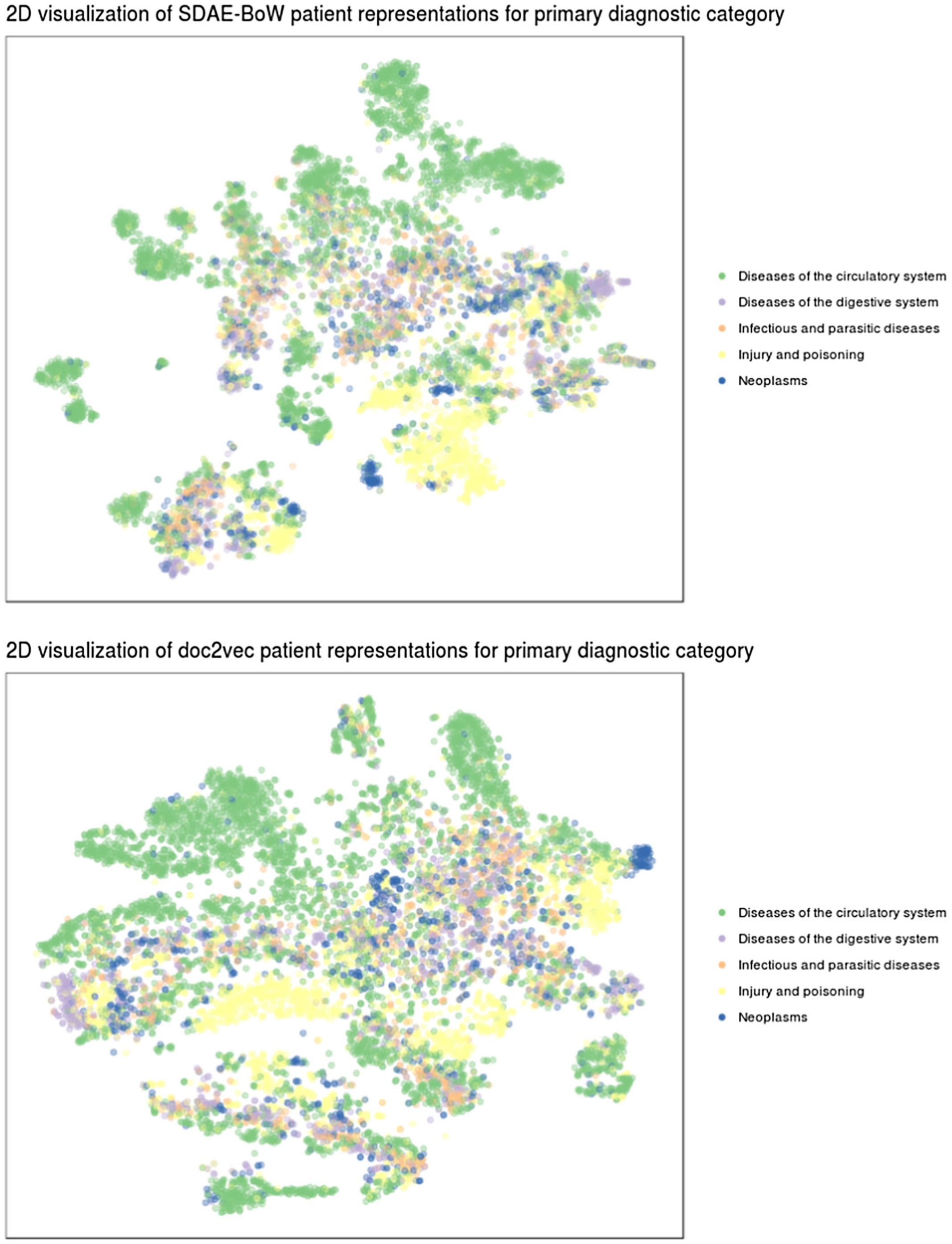

Fig. 5. t-SNE visualization of SDAE-BoW and doc2vec representations.

behavior of statistical models are transferable to different architectures and facilitate further research in this crucial direction.

Conflict of interest

The authors report no potential conflicts of interest.

Acknowledgements

This research was carried out within the Accumulate VLAIO SBO project, funded by the government agency Flanders Innovation & Entrepreneurship (VLAIO) [Grant No. 150056]. We thank Riccardo Miotto, Paulina Grnarova, and Florian Schmidt for sharing their model implementation details from related papers, and answering several questions about the same. Further, we thank Giovanni Cassani, Stéphan Tulkens, and Ella Roelant for their help with statistical significance analyses. We would also like to thank the anonymous reviewers for their useful comments.

Appendix A. Model hyperparameters

See Tables A.1 and A.2

Table A.1
Hyperparameters for stacked denoising autoencoder to learn dense patient representations, obtained after a randomized search. The default learning rate of 0.001 is used.

Feature set	Number of layers	Hidden dimensions	Dropout proportion
Bag-of-words	1	800	0.05
Bag-of-concepts	1	300	0.4

Table A.2
Hyperparameters for feedforward neural network classifiers for different tasks and feature sets, obtained after a randomized search. The default learning rate of 0.001 is used.

Task	Feature set	Number of layers	Hidden dimensions	Activation function
In_hosp	BoW	7	980	sigmoid
	SDAE-BoW	7	160	relu
	doc2vec	10	410	sigmoid
	[doc2vec, SDAE-BoW]	7	340	tanh
	BoCUI	3	680	sigmoid
	SDAE-BoCUI	3	560	sigmoid
30_days	BoW	10	220	relu
	SDAE-BoW	3	820	sigmoid
	doc2vec	2	900	sigmoid
	[doc2vec, SDAE-BoW]	8	430	sigmoid
	BoCUI	7	510	tanh
	SDAE-BoCUI	3	750	sigmoid
1_year	BoW	1	650	sigmoid
	SDAE-BoW	10	570	sigmoid
	doc2vec	3	1000	sigmoid
	[doc2vec, SDAE-BoW]	5	920	sigmoid
	BoCUI	1	290	sigmoid
	SDAE-BoCUI	6	290	relu
Pri_diag_cat	BoW	4	100	sigmoid
	SDAE-BoW	2	110	sigmoid
	doc2vec	9	600	relu
	[doc2vec, SDAE-BoW]	8	700	relu
	BoCUI	4	80	sigmoid
	SDAE-BoCUI	8	230	relu
Pri_proc_cat	BoW	2	220	sigmoid
	SDAE-BoW	5	890	relu
	doc2vec	3	980	relu
	[doc2vec, SDAE-BoW]	8	520	relu
	BoCUI	10	760	relu
	SDAE-BoCUI	6	540	relu
Gender	BoW	0	NA	NA
	SDAE-BoW	8	160	relu
	doc2vec	0	NA	NA
	[doc2vec, SDAE-BoW]	7	280	sigmoid
	BoCUI	5	410	relu
	SDAE-BoCUI	1	210	relu

References

[1] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.

[2] M. Baroni, G. Dinu, G. Kruszewski, Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22–27, 2014, Baltimore, MD, USA, vol. 1, Long Papers, 2014, pp. 238–247.

[3] F. Liu, J. Chen, A. Jagannatha, H. Yu, Learning for Biomedical Information Extraction: Methodological Review of Recent Advances, CoRR abs/1606.07993, 2016.

[4] N. Garcelon, A. Neuraz, V. Benoit, R. Salomon, S. Kracker, F. Suarez, N. Bahi-Buisson, S. Hadj-Rabia, A. Fischer, A. Munnich, et al., Finding patients using similarity measures in a rare diseases-oriented clinical data warehouse: Dr. Warehouse and the needle in the needle stack, J. Biomed. Inform. 73 (2017) 51–61.

[5] L. Li, W.-Y. Cheng, B.S. Glicksberg, O. Gottesman, R. Tamler, R. Chen, E.P. Bottinger, J.T. Dudley, Identification of type 2 diabetes subgroups through topological analysis of patient similarity, Sci. Transl. Med. 7 (311) (2015) 311ra174.

[6] P. Zhang, F. Wang, J. Hu, R. Sorrentino, Towards personalized medicine: leveraging patient similarity and drug similarity analytics, AMIA Summ. Transl. Sci. Proc. 2014 (2014) 132.

[7] Y. Wang, Y. Tian, L.-L. Tian, Y.-M. Qian, J.-S. Li, An electronic medical record system with treatment recommendations based on patient similarity, J. Med. Syst. 39 (5) (2015) 55.

[8] A. Gottlieb, G.Y. Stein, E. Ruppin, R.B. Altman, R. Sharan, A method for inferring medical diagnoses from patient similarities, BMC Med. 11 (1) (2013) 194.

[9] F. Wang, J. Hu, J. Sun, Medical prognosis based on patient similarity and expert feedback, 2012 21st International Conference on Pattern Recognition (ICPR), IEEE, 2012, pp. 1799–1802.

[10] K. Ng, J. Sun, J. Hu, F. Wang, Personalized predictive modeling and risk factor identification using patient similarity, AMIA Summ. Transl. Sci. Proc. 2015 (2015) 132.

[11] D.A. Lindberg, B.L. Humphreys, A.T. McCray, The Unified Medical Language System, Meth. Inform. Med. 32 (04) (1993) 281–291.

[12] A.E. Johnson, T.J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L.A. Celi, R.G. Mark, MIMIC-III, A Freely accessible critical care database, Sci. Data 3 (2016).

[13] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (Dec) (2010) 3371–3408.

[14] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: Proceedings of the 31st International Conference on Machine Learning (ICML-14), 2014, pp. 1188–1196.

[15] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, N. Elhadad, Intelligible models for healthcare: predicting pneumonia risk and hospital 30-day readmission, Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 1721–1730.

[16] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781, 2013a.

[17] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013b, pp. 3111–3119.

[18] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.

[19] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, TACL 5 (2017) 135–146.

[20] H. Larochelle, S. Lauly, A neural autoregressive topic model, in: Advances in Neural Information Processing Systems, 2012, pp. 2708–2716.

[21] R. Miotto, L. Li, B.A. Kidd, J.T. Dudley, Deep patient: an unsupervised representation to predict the future of patients from the electronic health records, Sci. Rep. 6 (2016) 26094.

[22] S. Dubois, D.C. Kale, N. Shah, K. Jung, Learning Effective Representations from Clinical Notes, arXiv preprint arXiv:1705.07025, 2017.

[23] H. Suresh, P. Szolovits, M. Ghassemi, The Use of Autoencoders for Discovering Patient Phenotypes, Workshop on Machine Learning for Health, NIPS, 2016, arXiv preprint arXiv:1703.07004.

[24] A.E. Johnson, T.J. Pollard, R.G. Mark, Reproducibility in critical care: a mortality prediction case study, in: Proceedings of Machine Learning for Healthcare, vol. 68, , JMLR W&C Track, 2017.

[25] M. Ghassemi, T. Naumann, F. Doshi-Velez, N. Brimmer, R. Joshi, A. Rumshisky, P. Szolovits, Unfolding physiological state: Mortality modelling in intensive care units, Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, pp. 75–84.

[26] P. Grnarova, F. Schmidt, S.L. Hyland, C. Eickhoff, Neural Document Embeddings for Intensive Care Patient Mortality Prediction, Workshop on Machine Learning for Health, NIPS, 2016, arXiv preprint arXiv:1612.00467.

[27] Y. Jo, L. Lee, S. Palaskar, Combining LSTM and Latent Topic Modeling for Mortality Prediction, 2017, arXiv preprint arXiv:1709.02842.

[28] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (Jan) (2003) 993–1022.

[29] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[30] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (Feb) (2012) 281–305.

[31] T. Tieleman, G. Hinton, Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude, COURSERA: Neural networks for machine learning, vol. 4(2), 2012, pp. 26–31.

[32] D. Erhan, Y. Bengio, A. Courville, P. Vincent, Visualizing Higher-Layer Features of a Deep Network, Tech. Rep. 1341, University of Montreal, also presented at the ICML 2009 Workshop on Learning Feature Hierarchies, Montréal, Canada., 2009.

[33] J. Li, X. Chen, E.H. Hovy, D. Jurafsky, Visualizing and Understanding Neural Models in NLP, in: NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12–17, 2016, pp. 681–691.

[34] J. Li, W. Monroe, D. Jurafsky, Understanding Neural Networks through Representation Erasure, 2016, CoRR abs/1612.08220.

[35] H. Suresh, N. Hunt, A.E.W. Johnson, L.A. Celi, P. Szolovits, M. Ghassemi, Clinical intervention prediction and understanding using deep networks, in: Proceedings of Machine Learning for Healthcare, vol. 68, JMLR W&C Track, 2017, arXiv preprint abs/1705.08498.

[36] D. Alvarez-Melis, T.S. Jaakkola, A causal framework for explaining the predictions of black-box sequence-to-sequence models, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9–11, 2017, pp. 412–421.

[37] D. Bahdanau, K. Cho, Y. Bengio, Neural Machine Translation by Jointly Learning to Align and Translate, 2014, CoRR abs/1409.0473.

[38] K.M. Hermann, T. Kociský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, P. Blunsom, Teaching Machines to Read and Comprehend, in: Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7–12, 2015, Montreal, Quebec, Canada, 2015, pp. 1693–1701.

[39] Z. Yang, D. Yang, C. Dyer, X. He, A.J. Smola, E.H. Hovy, Hierarchical Attention Networks for Document Classification, in: NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12–17, 2016, pp. 1480–1489.

[40] E. Choi, M.T. Bahadori, J. Sun, J. Kulas, A. Schuetz, W. Stewart, RETAIN: An interpretable predictive model for healthcare using reverse time attention mechanism, in: Advances in Neural Information Processing Systems, 2016, pp. 3504–3512.

[41] A. Engelbrecht, I. Cloete, Feature extraction from feedforward neural networks using sensitivity analysis, in: Proceedings of the International Conference on Systems, Signals, Control, Computers, 1998, pp. 221–225.

[42] Y. Dimopoulos, P. Bourret, S. Lek, Use of some sensitivity criteria for choosing networks with good generalization ability, Neural Process. Lett. 2 (6) (1995) 1–4.

[43] M. Gevrey, I. Dimopoulos, S. Lek, Review and comparison of methods to study the contribution of variables in artificial neural network models, Ecol. Modell. 160 (3) (2003) 249–264.

[44] M. Van Gompel, K. van der Sloot, A. van den Bosch, Ucto: Unicode Tokeniser, Tech. Rep., Tilburg Centre for Cognition and Communication, Tilburg University and Radboud Centre for Language Studies, Radboud University Nijmegen, 2012.

[45] E. Soysal, J. Wang, M. Jiang, Y. Wu, S. Pakhomov, H. Liu, H. Xu, CLAMP — a toolkit for efficiently building customized clinical natural language processing pipelines, J. Am. Med. Inform. Assoc. (2017), http://dx.doi.org/10.1093/jamia/ocx132.

[46] Ö. Uzuner, B.R. South, S. Shen, S.L. DuVall, 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text, J. Am. Med. Inform. Assoc. 18 (5) (2011) 552–556.

[47] E. Scheurwegs, M. Sushil, S. Tulkens, W. Daelemans, K. Luyckx, Counting trees in Random Forests:predicting symptom severity in psychiatric intake reports, J. Biomed. Inform. 75 (2017) ISSN 1532-0464.

[48] World Health Organization, International Statistical Classification of Diseases and Related Health Problems, vol. 1, World Health Organization, 2004.

[49] J. Cohen, A coefficient of agreement for nominal scales, Educ. Psychol. Measure. 20 (1) (1960) 37–46.

[50] E.W. Noreen, Computer-intensive Methods for Testing Hypotheses, Wiley, New York, 1989.

[51] S. Kokoska, D. Zwillinger, CRC standard probability and statistics tables and formulae (pp. Section 14.7), Chapman & Hall, CRC, Boca Raton, Fla., 2000.

[52] L.v.d. Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (Nov) (2008) 2579–2605.