Infer Cause of Death for Population Health Using Convolutional Neural Network

Hang Wu
Georgia Institute of Technology and Emory University
U.A. Whitaker Building, Suite 4236
Atlanta, GA 30332
hangwu@gatech.edu

May D. Wang
Georgia Institute of Technology and Emory University
U.A. Whitaker Building, Suite 4236
Atlanta, GA 30332
maywang@gatech.edu

ABSTRACT

In biomedical data analysis, inferring the cause of death is a challenging and important task, which is useful for both public health reporting purposes, as well as improving patients' quality of care by identifying severer conditions. Causal inference, however, is notoriously difficult. Traditional causal inference mainly relies on analyzing data collected from experiment of specific design, which is expensive, and limited to a certain disease cohort, making the approach less generalizable.

In our paper, we adopt a novel data-driven perspective to analyze and improve the death reporting process, to assist physicians identify the single underlying cause of death. To achieve this, we build state-of-the-art deep learning models, convolution neural network (CNN), and achieve around 75% accuracy in predicting the single underlying cause of death from a list of relevant medical conditions. We also provide interpretations for the black-box neural network models, so that death reporting physicians can apply the model with better understanding of the model.

CCS CONCEPTS

• **Computing methodologies** → **Causal reasoning and diagnostics**; **Neural networks**;

KEYWORDS

Causal Inference, Deep Learning, Interpretability

ACM Reference format:
Hang Wu and May D. Wang. 2017. Infer Cause of Death for Population Health Using Convolutional Neural Network. In *Proceedings of ACM-BCB'17, August 20-23, 2017, Boston, MA, USA.*, , 10 pages.
DOI: http://dx.doi.org/10.1145/3107411.3107447

1 INTRODUCTION

Physicians and medical examiners are faced with the challenges of inferring causes of death in their death reporting routine. When a death case occurs in hospital, for example, a physician will file a death certificate to the state agency, summarizing the demographics, a sequence of up to 20 medical conditions relevant to the death, coded using ICD-10 standards, and a single underlying cause of death the physician thinks most probable. These mortality data are finally aggregated and recorded by the national vital statistics system of the national center for health statistics (NCHS).

With more and more such death certificates available, it is natural for us to wonder: can we utilize the large-scale observational datasets, to build a causal inference model that can identify the cause of death based on the observations?

Causal inference, which aims to uncover the mechanism behind observations or predict the effect of an intervention to the system, is an important task in biomedical data analysis. Identifying the causes of diseases and deaths will facilitate public health reporting process, and improve individual's quality of care by guiding design of treatment for diseases.

Such causal inference tasks are notoriously difficult, as for each patient, we can only observe one of all the potential outcomes, and to determine the causal effect, we need to compare the observed ones against all others with some approximation schemes. Several biomedical studies have studied the causal structures inside patients with one type of disease, and design random or non-random experiments to analyze the causal effects. These studies provide great insight into the diseases they studied, but there are certain limitations to it: collecting experimental data is expensive and time-consuming; moreover, in a general hospital setting, physicians and nurses might be dealing with a combination of complicated medical conditions, and need to prioritize the treatment and resource allocation, which requires better understanding of the effect of combination of medical conditions, and current studies on one type of disease might be insufficient.

In this study, we take a novel approach, to study death certificates data and build predictive model to assist physicians in identifying the single underlying cause of death.

To predict the single cause of death, given the input as a sequence of conditions, poses several challenges: 1) The input sequence of conditions is highly unstructured, so if we are to use traditional one-hot vector to represent each of the conditions, we will be dealing with large dimension of inputs, and succeeding feature extraction will require large computation resources. 2) Since essentially we are selecting one medical condition out of a list of conditions, and each death certificate will be different, we need to seek a model that can adaptively process inputs of different lengths, and predict accordingly.

To address such challenges, inspired by the recent success of deep learning, we propose to apply deep learning, specifically convolutional neural networks, to build our causal inference model. Deep learning has shown great performance in processing raw data

ACM-BCB'17, August 20-23, 2017, Boston, MA, USA.

DOI: http://dx.doi.org/10.1145/3107411.3107447

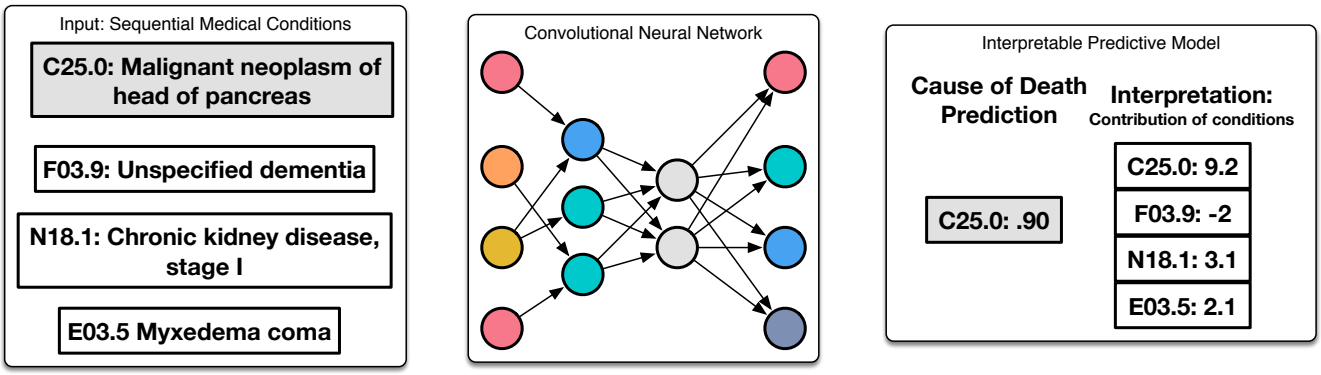

Figure 1: *Overall Architecture.* In this paper, we analyze a causal inference task originating from the death reporting process. In the process, for each of the death case, the physician will select a sequence of medical conditions, coded in ICD-10 standard, that is most relevant to death, and identify one condition as the underlying cause of death. To predict the cause of death using the sequence as input, we build a convolutional neural network, and provide interpretations of the model's prediction for physicians.

in various formats and obtain trainable representation specific to the tasks, and deep learning structures can be configured to adapt to inputs of different shapes [8].

Our model will facilitate the further analysis of causality modeling in several aspects: 1) although our current model is build on NCHS mortality data, it is directly transferrable to analyzing more specific electronic health records data (EHR) during hospital admission. By parsing EHR to a sequence of observed conditions format, we can identify the most likely cause of death, which can help physicians attend more to important conditions, thus improving the quality of care. 2) another advantage with deep learning models lie in the distributed representations of the input medical conditions, and such representation is universal and can be applied to other data analysis tasks to understand the relationships between them better.

The overall model structure is presented in Fig. 1.

Our contribution in this paper is mainly two-fold:

- We designed a convolutional neural network (CNN) model to identify the underlying cause of death from a list of relevant medical conditions using death certificates data, and achieved about 75% accuracy, a significant performance improvement over the conventional methods.
- We illustrated how we can interpret the black-box deep learning models, so that physicians can understand and choose whether to adopt the machine prediction.

In the rest of the paper, we first review related work on cause of death, causal inference, and deep learning models. We present our model in Section 3, with experiment results in Section 4. We then show how we can interpret the black-box algorithms, followed by discussions and conclusions.

2 RELATED WORK

2.1 Causes of Death

Understanding causes of death has been a big challenge in biomedical research, as discerning new risk factors for death can help physicians understand the mechanism of death and certain disease, so to improve the quality of care for patients.

Previous research mainly focuses on discovering the cause of death for patients with specific disease types, or discovering new factors that can be identified as cause of death. The Emerging Risk Factors Collaboration studied the risk factors associated with diabetes [4] and Khorana et al. studied for the cancer patients with chemotherapy [16]. Other studies include causes of death for neonatal deaths [22], Alzheimer's death [30], and sclerosis [43].

Back in 1986, Israel et al [13] first pointed out the potential multiple cause-of-death data, and with the popularity of statistical data analysis methods, the dataset gained more popularity in the healthcare research community. Redelings et al. [36] analyzed the associations of cause-of-death conditions, and Jiang et al. [15] analyzed the evolution of causes with a topic model approach. Some researchers also look in to the subgroups of patients associated with certain diseases, for example, McCoy et al. on asthma [26], Melamed et al. on sepsis [27]. Yet these studies mostly take on an association analysis approach and few have focused on the problem of causal identification.

2.2 Causal Inference

Causal inference plays a vital role in analyzing biomedical observational study, as it can help determine the treatment effect of certain drugs or procedures [19, 42]. Rubin et al. [39] was the first to analyze the causal inference problem in such experiment problems, and the paper mainly discussed the difference between random experiments and non-random experiments. Such experiments aim to discover the effect of a binary treatment, and the term "random/ non-random" refers to whether the assignment of treatment depends on the patients' conditions or not. In a random design experiment, since the treatment is assigned randomly, it's straightforward to analyze the treatment effect by comparing populations with treatment to populations without treatment. However, most of biomedical experiments are non-random observational studies,

so it's crucial to identify the causal structure, or design special algorithms to account for the cofounding effects in such studies.

Identify causal structure mainly builds on the causal graph framework proposed by Pearl et al. [34], where variables/ features are represented as nodes, and edges indicate causal relationships. Under this framework, when the causal structure is specified beforehand, often by domain experts, structural equation model (SEM) [2, 32] can analyze the causal effects of intervention on certain variables.

Oftentimes, such structure is unknown or incomplete, thus, a series of work is conducted to learn the causal structure. Constraint-based algorithm, such as PC algorithm [44], learns the graph by exploiting conditional dependence relationship between variables. Score-based method [3] designs an evaluation metric to score each potential causal structure, and finds the one with highest score.

On the other hand, to analyze the treatment effect from observation studies, we could also adopt the potential outcomes framework [40]. Two popular approaches are matching [31, 41, 46] and propensity score. By matching, we find a pair of two instances, that receive opposite treatment, while are most similar in other features. In this way, the difference between them after treatment can be used as an estimate for the treatment effect. Propensity score works by reweighing instances to convert an observation study to a pseudo-random experiment and work with random experiment [1, 38].

2.3 Deep Learning and CNN

The past decade has witnessed the success of deep learning, enabled by effective training algorithm [9, 18], high performance computing structure including GPU, and large-scale labeled datasets [5]. Deep learning has shown great capabilities in image classification [20], image segmentation[24], text analysis [48], and reinforcement learning [29].

Recently, biomedical researchers are applying deep learning to biomedical data analysis [10]. Liu et al. analyzed brain imaging with deep learning, to achieve early diagnosis of Alzheimer's disease; Esteva et al. used deep learning to classify a skin cancer dataset containing about 130,000 images and beat human physician accuracy[7]; Suo et al. [47] applied deep belief nets to derive risk factors from electronic health records.

CNN was originally proposed to address image classification [23], initially dealing digit recognition. It revived as the standard practice of image classification around 2012, with the success of AlexNet beating human accuracy. Then CNN showed great success in other image processing tasks, such as image segmentation [24] and human action recognition [14]. CNN then was also applied to sentence classification [17] and inspired several follow-ups [6, 11]. Another direction of work uses recurrent neural network [21], and showed better capabilities in text generation [48], and text conversion to other modalities [25].

3 MODEL

3.1 Problem Formulation

In the death reporting scenario, a physician will first select several conditions, coded by ICD-10 codes [1], as the conditions most relevant to death. The conditions are then recorded as they are sequential observed. Among these conditions, one of them will be predicted as the cause of death, based on physicians' expertise and the understanding of the death case.

In this paper, we are interested in automating the latter part of the process, identifying the one cause of death from the list of conditions. Mathematically, suppose we have a vocabulary of ICD-10 conditions V, with the total number denoted as $|V|$. We are given a dataset of $\{x_i, y_i\}, i = 1, ..., N$, where $x_i = [c_{i,1}, ..., c_{i,i_k}]$ is the sequence of i_k relevant conditions recorded, and y_i is the identified cause of death. We are interested in learning a classifier as $f(x_i) = y_i \in V$, essentially a multi-class classification problem, with input being a list of items also from the vocabulary V.

The sequential and discrete nature of the ICD-10 conditions in our case presents a strong analogy with natural language. We can regard each ICD-10 condition as a word, and physicians use a sequence of ICD-10 conditions to describe a death case, which is similar to a human sentence describing a concept. So our problem can be considered as a sentence classification problem. However, we should note a profound difference between our problem and sentence classification: traditional sentence classification generally deals with binary classification (e.g., positive sentiment vs. negative sentiment), or a few classes that indicate the topic of the sentence (e.g., religious, movie, news, etc.). In our cause of death identification, we are dealing with thousands of ICD-10 vocabulary as the final class label, which means the final classification probability could be sparsely distributed among the vocabulary, posing a computational challenge.

In light of the recent success in deep learning in text classification cases, we adopt the convolutional neural network (CNN) framework to our application, and made necessary modifications.

3.2 Convolutional neural network(CNN) for sentence classification

Applying CNN to sentence classification is proposed by [17], and has thus been extensively studied and applied in the literature. Here, we first review the basics of the CNN framework, and then introduce the model we modified.

For a sequence $x = [c_1, ..., c_k]$, where each condition $c_j \in \mathbb{R}^{|V|}$ is one-hot vector encoding representation, we first apply a word embedding to obtain a distributed representation for it [28] in a lower dimensional space $\mathbb{R}^D$. Equivalently, we are learning a weight matrix $W \in \mathbb{R}^{|V| \times D}$, so that we embed each condition c_j as the jth row of the matrix, with the simple matrix multiplication $v_j = c_j W$.

After the embedding, we concatenate all the embedding vectors, and obtain the initial representation for the sequence as

$$v = v_1 \oplus v_2 \oplus ... v_k$$

. The convolution operator is applied to a segment of the sequence, determined by the window size. For example, for a window of size of H, the convolution uses a filter $m \in \mathbb{R}^{H \times D}$, and the result of this convolution to a segment $v_{i:i+H-1} = [v_i, v_{i+1}, ..., v_{i+H-1}]$ is

$$z_i = f(m * v_{i:i+H} + b_0)$$

, where $*$ is the convolution operator, b_0 a scalar bias term, and f a nonlinear transformation, such as Tanh and ReLU. The rationale for a convolution operation lies in that conditions that occur closely in

[1] Starting FROM 1999, NCHS has switched from ICD-9 to ICD-10 coding system.

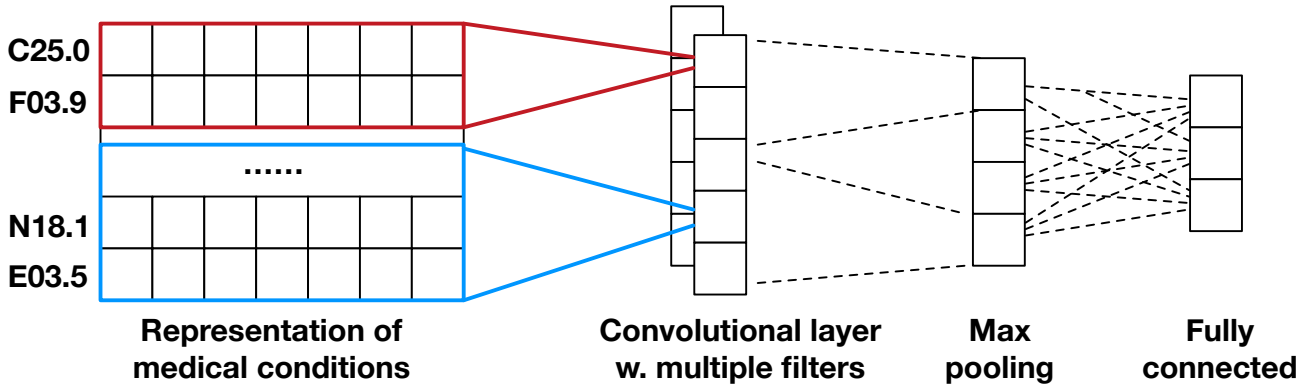

Figure 2: *Model architecture.* The convolution neural network architecture we used in the project is from Kim et al. [17], which contains a 1D convolution layer and a max pooling, followed by a fully connected layer as logit outputs.

the sequence should share some characteristics and be correlated to each other, as we would expect in the 2D image case.

We can further apply a pooling operation to the obtained feature map $[z_1, ..., z_k]$, which finds the maximum or average among all the z_js. The intuition is that for each filter, we find the most important features, and this operation naturally handles variable length sequences.

In practice, we can apply several convolution filters to obtain several corresponding features. These intermediate features can be passed again to convolution filters then nonlinear transformation, and the stacking of these layers compose a deep neural network structure, which some describes as Parrellel-CNN.

In the ultimate layer, for the hidden feature vector $u \in \mathbb{R}^{D_1}$, we apply a fully connected layer to obtain the final output $y = g(u^T Q + b_1)$. The parameter $Q \in \mathbb{R}^{D_1 * |V|}$ maps the hidden feature to a distribution over all the vocabulary V, and the condition with maximum probability is predicted as the cause of death.

An illustration of the model architecture is shown in Fig. 2.

3.3 Proposed method: convolutional neural network with dynamic computation graph

A challenge of the above vanilla CNN structure lies in the final parameter matrix Q, as computing a soft-max distribution over the whole vocabulary of codes can result in sparse entries, and makes the inference more difficult.

To overcome such challenge, we note that instead of predicting over the whole vocabulary, we only need to select one condition out of the input sequence, which has a length significantly smaller than the total vocabulary. Moreover, each of the sentence has a potential different length, so this requires that we dynamically build a neural network for each of the input.

The new "Define by Run" paradigm of deep learning framework has enabled us to build such dynamic neural networks. In brief, instead of specifying a static network structures before any input is fed, now we take a particular training sample, a sequence in our case, and define the network structure from this input to its output. This is also the same for testing phase, where we dynamically construct a network architecture for each test sample, potentially of various shapes. Although this practice seems more tedious in terms of computation, it supports dynamically sized data, such as our sequences of medical conditions with variable lengths, and also reduces the complexity of computation in computation graph implementation.

3.4 Regularization

Deep neural networks tend to overfit the data, thus, in our implementation, we imposed following three regularization techniques

Batch Normalization (BN). Batch Normalization (BN) [12] works on the intuition that during the training of neural network, because of the changes of the network parameters, the distribution of the nonlinearity activation function also shifts accordingly. Mathematically, for each dimension x_k of the D dimension input feature $x = [x_1, ..., x_D]$, the new normalized feature is

$$\hat{x}_k = \frac{x_k - \mathbb{E}[x_k]}{\sqrt{Var[x_k]}}$$

, where the expectation and variance are computed for each mini-batches.

Reducing such shift in covariates can accelerate the convergence in training, and sometimes has the benefit as regularization.

Dropout (DO). Dropout [45] is one of the most successful regularization techniques. In brief, the weights of a proportion of the hidden neurons in the network layers are chosen to be 0s during training. For example, in the ultimate layer of our network, we have the final output computed as $y = g(u^T Q + b_1)$. Instead of applying Q directly, we randomly generated as mask matrix M_Q, where each entry of it is a Bernoulli variable with specified probability p in $(0, 1)$, often set to 0.5 as suggested by literature [17, 45].

The output is then

$$\hat{y} = g(u^T (Q. \times M_Q) + b_1)$$

, where $.\times$ denotes the element-wise multiplication.

During test, we scale the learned weight matrices by p as $\hat{Q} = pQ$, and use them without dropout to predict unseen sentences.

Early Stopping (ES). Early stopping is also widely used as a regularization technique for almost any types of machine classifiers, that rely on sub-gradient methods as training algorithms. It attracts wide popularity in training deep neural networks as it can help save a great amount of computation time while preserving considerable test performance.

We first partition a small proportion of the training set as our development set, and train classifiers on the rest of the training set. Once we observe the test performance on development set is worse than the training performance on the rest, and the test performance exceeds a threshold we set beforehand, we can conclude that the training might already overfit the data, and terminate the training process.

3.5 Discussion: CNN Vs. Bag of Words

Traditional sentence processing mainly use the bag-of-n−grams method to represent the sentence [50], where each dimension is the term frequency times inverse document requency (tf-idf) for the n gram for the sentence. With features extracted, off-the-shelf classification methods, such as naive Bayes, support vector machine, can be applied to classify the sentences.

Despite its simplicity, there are several disadvantages of this approach compared to CNN: 1) the n -gram tend to be really sparse, and even infeasible to compute, when the n is large; 2) the one-hot representation of these n−grams ignore the shared parts between all n−grams, as well as the distributed representations obtained by embedding words; 3) training such models requires first loading all data into RAM to process the TFIDF matrix, so we cannot adopt mini-batch training techniques here, making the model less scalable.

4 EXPERIMENTS & RESULTS

4.1 Dataset overview

For the experiments, we pick the death certificates in the United States from Year 2014, which contains approximately 2 million records of death cases [33]. After preprocessing, removing identical records and filter out records with length less than 3, we obtain 1,499,128 records.

The ICD-10 codes, in the format of A123.4, observe a hierarchy structure, where the digits before the dot can be regarded a coarse high-level classification of the condition.To save computation, we here use the coarse version, and as a result, we obtain a vocabulary of input conditions of 1610 and a total of 1180 possible classes as causes of death.

4.2 Configurations for CNN

For our method, we experiment both static and dynamic constructed CNN. The static constructed one is referred to as *CNN-static*. As for the dynamic version, we can either train the network as a static structure, and when testing, only selects ones that are present in the input, or we can use a dynamic structure in both training and testing phase. These two versions are referred to *CNN-dyn-eval* and *CNN-dynamic* respectively.

The network structure we used is specified as follows:

```
Input -> Embedding Layer -> Convolution & Pooling
-> Batch Normalization -> Dropout
-> Fully Connected Layer -> Output
```

The embedding dimension is set to 128, and three kernel sizes for the convolution layers are 3,5,7. Dropout probability is set to .5 as suggested in [17] and maximum norm of parameters are set to 3.0. Our model is built with PyTorch [8], and adapted from open source implementations [2].

4.3 Baseline methods

For the baselines, we implemented two types of baseline algorithms: traditional BoW classification, and shallow classifiers built on embeddings.

For the bag-of-words feature extraction, we first construct a count matrix X, where $X[i, j]$ denotes the count of word j appearing in the document i. Then tf-idf transform, short for "term frequency times inverse document frequency" is applied to X to obtain the final feature matrix. We used bag-of-words, instead of more expressive n-gram, because it would require much more than the computation power we have available right now. We then apply naive Bayes (NB), support vector machine (SVM), and logistic regression (RF) to the feature matrix, and obtain predictions.

For shallow classifiers, we use the architectures shown below

```
Input -> Embedding Layer -> Vector Averaging
-> Fully Connected Layer -> Output
```

After we embed all the medical conditions of a sequence, we average them as the vector representation as the sentence, then use a fully connected layer to obtain the final output. We use cross entropy loss (equivalent to a logistic regression model), and multi margin loss (equivalent to a support vector machine model), and also implemented the three variants w./o. dynamic graph in PyTorch.

4.4 Experiment Settings

We randomly partition the data into training, development, and test sets with ratio 7.9:.1:1. The hyper-parameters were selected based on the performance on the development set and then tested on the test set. We also reported the model performance with and without early stopping using development set.

In naive Bayes, we don't have a parameter to tune, and we tune the regularization parameter for SVM, and the number of trees for RF. As for CNN, we mainly tune the kernel sizes. The results were reported by averaging 3 runs of experiments.

BoW classifiers are run in CPU with 60 GB RAM and we used implementation of this pipeline from Scikit-Learn [35]. CNN and shallow learners are trained with NVIDIA K80 GPU. We use a mini-batch of 64 for training, and we set the maximum number of epochs to 2 (the number of iterations over the whole training data). We used Adam with adaptive learning rate as the sub-gradient optimization method. Both training with and without early stopping are tested. Training such networks averaged to about 5 hours under our configurations, while testing a single case using a trained model takes about seconds.

[2] https://github.com/Shawn1993/cnn-text-classification-pytorch

Table 1: Classification Results

Classifier Name	Test Loss	Test Accuracy	Test Micro F1	Test Cohen Kappa
CNN-static	0.799±0.009	75.481±0.345	8.4e-06±3.8e-08	8.3e-06±3.8e-08
CNN-static-es	0.902±0.017	73.681±0.346	8.2e-06±3.8e-08	8.1e-06±4.0e-08
CNN-dyn	3.996±0.861	68.261±0.362	7.6e-06±4.0e-08	7.5e-06±3.9e-08
CNN-dyn-es	3.765±0.400	51.946±1.130	5.8e-06±1.3e-07	5.6e-06±1.2e-07
CNN-dyn-eval	**0.738±0.007**	**75.787±0.179**	**8.4e-06±2.0e-08**	**8.3e-06±1.8e-08**
CNN-dyn-eval-es	0.826±0.044	73.184±1.254	8.1e-06±1.4e-07	8.0e-06±1.4e-07
LR	1.011±0.007	66.762±0.298	7.4e-06±3.3e-08	7.3e-06±2.9e-08
LR-es	1.007±0.024	66.706±0.681	7.4e-06±7.6e-08	7.3e-06±7.6e-08
LR-dyn	0.852±0.014	66.946±0.431	7.4e-06±4.8e-08	7.3e-06±4.9e-08
LR-dyn-es	0.886±0.015	66.326±0.720	7.4e-06±8.0e-08	7.2e-06±8.0e-08
LR-dyn-eval	0.842±0.012	67.295±0.514	7.5e-06±5.7e-08	7.4e-06±6.0e-08
LR-dyn-eval-es	0.881±0.006	66.633±0.297	7.4e-06±3.3e-08	7.3e-06±3.1e-08
SVM	1.285±0.056	64.597±1.042	7.2e-06±1.2e-07	7.0e-06±1.2e-07
SVM-es	2.039±1.116	62.812±2.312	7.0e-06±2.6e-07	6.8e-06±2.6e-07
SVM-dyn	14.588±1.857	44.984±1.771	5.0e-06±2.0e-07	4.8e-06±2.3e-07
SVM-dyn-es	13.007±2.572	45.068±2.111	5.0e-06±2.3e-07	4.9e-06±2.3e-07
SVM-dyn-eval	13.800±1.670	47.377±1.891	5.3e-06±2.1e-07	5.1e-06±2.4e-07
SVM-dyn-eval-es	11.253±2.384	48.081±2.345	5.3e-06±2.6e-07	5.2e-06±2.8e-07
LR-BoW	6.486±0.000	8.3±0.000	8.3e-02±0.0e+00	0.0e+00±0.0e+00
NB-BoW	4.864±0.000	8.3±0.000	8.3e-02±0.0e+00	0.0e+00±0.0e+00

4.5 Evaluation Metrics

To evaluate the performance of the algorithms, we adopt common evaluation metrics for multi-class classification.

Accuracy (ACC). Accuracy (ACC) measures the percentage of the sequences that are predicted correct.

Cross Entropy Loss. For a classifier with logits as output, the classification cross entropy loss is defined as

$$Loss(logit, class) = -logit[class] + log(\sum_{j=1}^{C} exp(logit[j]))$$

, where *class* is the true class of the sample, and *logit* is a vector containing the logits for all the classes.

F1. To account for the potential imbalance between false-negative and false-positives, F1 measure computes the harmonic average of precision and recall. F1 is in the range of $[0, 1]$ and the higher, the better predictive power. In the case of multi-class classification, we compute for each class an F1 measure, and then use the average of these F1s as the final metric.

Cohen's kappa. Cohen's kappa is a statistical that measures the inter-rater agreement between two classification output, defined as

$$\kappa = 1 - \frac{1 - p_o}{1 - p_e}$$

, where p_o is the accuracy we mentioned above, and p_e is the probability of agreement by chance. A perfect agreement will have $\kappa = 1$, and $\kappa < 0$ indicates a no agreement other than by chance.

4.6 Results & Discussion

We can see that Bag-of-Words classification techniques fail to capture the causal inference in our case, giving a poor classification performance. By examining the classification output, the classifier simply outputs the class that has the highest frequency, thus resulting in an identical output in all runs. This most likely results from the feature extraction process, where Bag-of-Words only consider the frequency of words.

Between the shallow classifiers and CNN, we can see that CNN obtained the highest classification accuracy, and the lowest loss with dynamic evaluation configuration, beating all other models with lowest variance.

Models with dynamic neural network structure in training and testing phase have a varied performance, and in the case of CNN and LR, it has the benefit of improving classification performance slightly, while in SVM, the benefit is not observed.

Early stopping as a regularization decrease the number of batches from about 21000 to about 9000 in all three models, saving almost

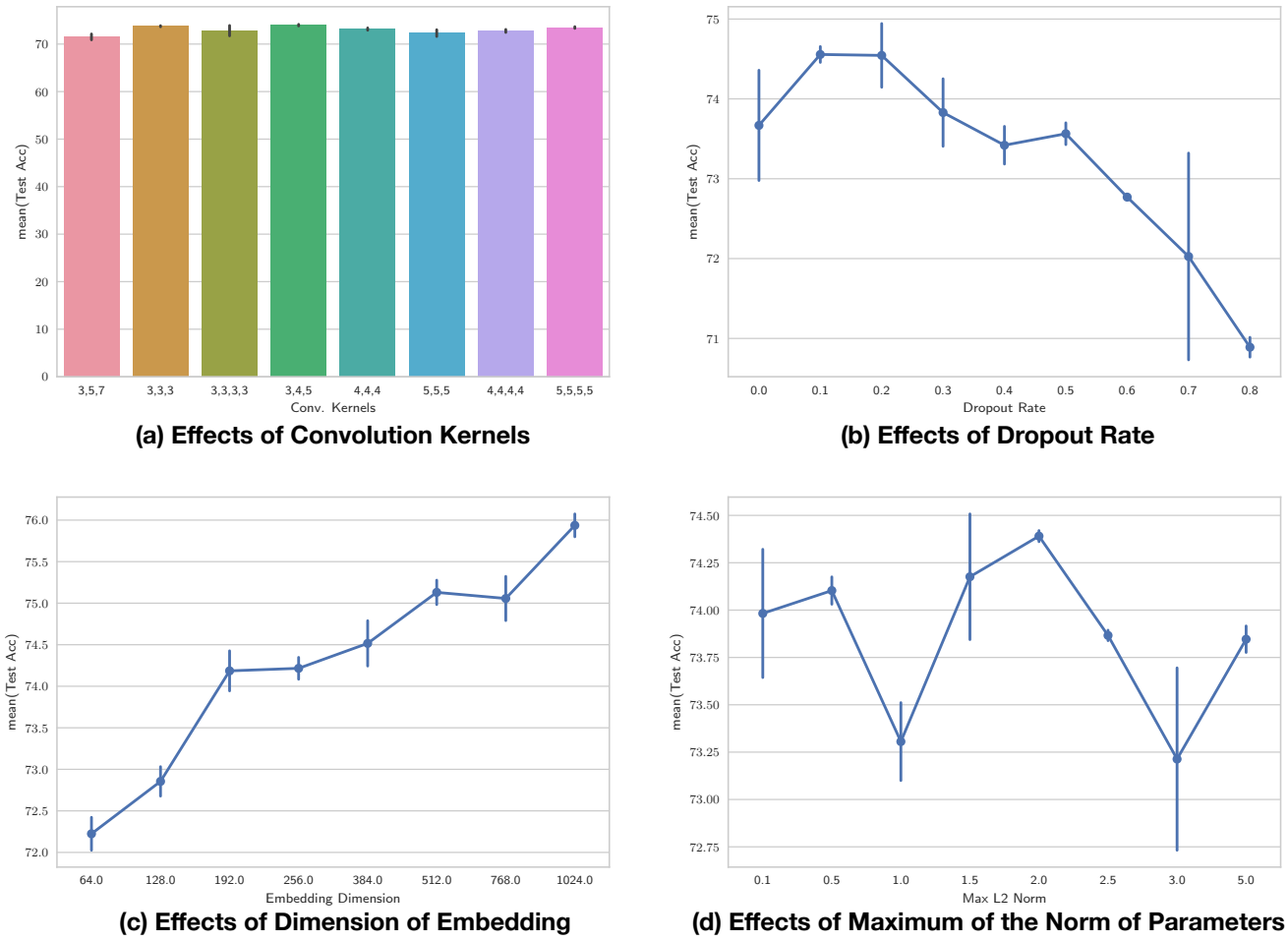

Figure 3: *Parameter Analysis.* We varied several key parameters of the CNN model and overall, the scale or the pattern of influence is not clearly shown from our experiments. The Y-axis is the accuracy on test sets with error bars and X-axis is the parameter we analyzed.

half of the running time, however, most of models with early stopping have a worse performance than the ones without early stopping, indicating that simply comparing development set training accuracy might not be a sufficient criterion for model overfitting, but just the oscillation behavior in the local minima region.

While some algorithms present performance in terms of accuracy and loss, all algorithms perform poorly evaluated with Cohen-Kappa's and microF1, mainly because of the fact that we are dealing with an extreme large number of classes. Take microF1 as an example, it's an average of the F1 value for each of the classes. For some of the classes, it may only have one or two samples, if the algorithm predicts these few samples wrong, its precision will be zero, thus a zero F1 score, significantly decreasing the final averaged microF1. In future, it may be of interest to design algorithms that can achieve high F1 in such classification case, as classification with extremely large number of classes itself is an interesting research question.

4.7 Parameter Analysis

Deep learning models have seemingly a large number of parameters to tune, in our case, such as the convolution kernels, the maximum norm, the dropout rate, and the dimension of embedding. Here we briefly show the effect of varying these parameters on the final prediction accuracy.

The base model is the standard static version of CNN, and we vary these four parameters, and plot the prediction accuracy on the test set with standard deviation as error bars.

From the figure, we can see that although there are several parameters to set, with a good model architecture, the specific values of these parameters don't influence the final outcome much. Except with the dimension of embedding medical conditions, we found that as the size of embedding dimension increases, the test performance slightly increases.

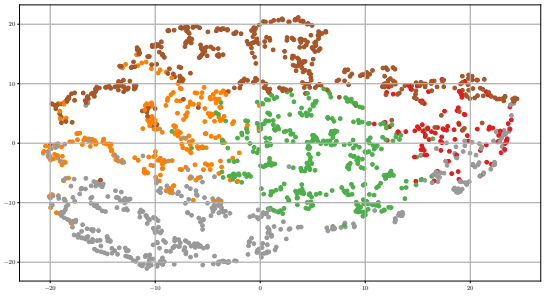

Figure 4: Embedding Visualization. After dimension reduction with kernel PCA, and t-SNE visualization, we plot the embeddings of medical conditions. We label the points with colors using the clustering results on the original embeddings. We can see the embedding vectors are scattered in the 2 dimensional data, and nicely clustered in five groups.

4.8 Analyzing Embeddings of Medical Conditions

One side-product of our model is the distributed representations of medical conditions, which we visually present in Fig. 4, using t-Distributed Stochastic Neighbor Embedding (t-SNE) [49] following a dimension reduction using kernal PCA to a three dimensional space.

The color of these dimension reduced points are obtained by running K-means clustering the embedding matrix of size (1610, 128), with number of clusters set to 5. We can see the embedding vectors are scattered in the 2 dimensional data, and nicely clustered in five groups.

To understand the embedding better, we also show some of the clustering results, with top conditions in each cluster 5. We conjecture the embedding of medical conditions reflect two characteristics, the likelihood of each condition causing death, and the physiological relationship between them, so we may not see a very clear pattern of the clustering results. For example, cluster 5 mainly consists of circulatory related conditions, while cluster 2 contains a few more severe diseases.

5 INTERPRETATION OF CAUSE IDENTIFICATION

Right now there are no golden standards as to explain the cause of death, and physicians mainly fill the death certificates speaking from their own perspective. The collection of death certificates from across the nation is thus a good resource to distill the knowledge in understanding cause of death, by training such supervised learning model. To help physicians really understand why our model gives predictions for causes given each sequences, we need to give proper interpretations for the black-box models, i.e., to provide relation between the input sequence and the final selected condition.

5.1 LIME-Local Interpretable Model-Agnostic Explanations

In this section, we briefly introduce the Local Interpretable Model-Agnostic Explanations model proposed [37], which can explain any type of black-box predication algorithms. To understand which parts of the input are contributing to the final prediction, the model perturbs the input around its neighbors, and analyzes the classifier's predictions on these perturbed instances with a sparse linear model. Then the weights from the linear model indicates how important the corresponding part is to the final predictions.

5.2 Case study

Here we showcase how the model will explain an unseen instance. we synthesize a patient history and the resulting sequence is

'I50,J44,I25,T82,Y83,I73,I10,J96,I64,K21,F17'

The explain model then outputs the top most likely cause of death as I25, as well as why certain conditions are more likely to cause the death, explained by the input conditions. We show such explanation in Fig. 6.

Because of the constraint of the data, we now can only pinpoint these ICD10 conditions, which admittedly, is still limited.

If we are to have a more complete dataset, where we have the complete history of patients, as well as the identified cause of death, we are then able to train a deep learning model predicting cause of death using the whole medical history. With the model and our explainer, we can understand in a greater detail about death cases. When a new patient is admitted to hospital, we can use such model to understand, which condition is the most likely cause of death, and which symptoms are contributing to this causality sequence.

Another interesting application of such interpretation model is that we can provide several predictive models for physicians, as well as their interpretations, and ask physicians to choose the one that matches human knowledge more. Running these tests will help us choose an accurate model that is more interpretable.

6 CONCLUSION & FUTURE WORK

In this paper, we showed how a modern deep learning architecture, CNN, can be adapted to identify the cause of death. The model shows significant improvement over the traditional baselines, and can handle even larger scale datasets than traditional methods. We also provide human understandable interpretation for the model, so that death reporting physicians.

The current work is limited by the dataset itself, and we are working

There are several ways our current work can be extended: First, we may deploy the model in a general EHR setting, where we can identify the most probable potential causes of death, so as to alert physicians and nurses to attend to more critical conditions. Second, there are medical ontologies specified by domain experts, which record all the viable causal relations between medical conditions. We can seek to integrate the guidance and constraint from these constraints into our models, and reach a model derived both from data and human knowledge. Moreover, it will be interesting to see how other deep learning architectures will perform in this task and other causal inference problems.

J69.X: Pneumonitis	J96.X: Respiratory failure, not elsewhere classified	I42.X: Cardiomyopathy	A41.X: Other sepsis
T42.X: Poisoning by, adverse effect of and underdosing of antiepileptic, sedative-hypnotic and antiparkinsonism drugs	K76.X: Other diseases of liver	G93.X: Other disorders of brain	I50.X: Heart failure
R26.X: Abnormalities of gait and mobility	G30.X: Alzheimer's disease	E83.X: Other disorders of fluid, electrolyte and acid-base balance	E88.X: Other and unspecified metabolic disorders
I69: Sequelae of cerebrovascular disease	T50.X: Poisoning by, adverse effect of and underdosing of diuretics and other and unspecified drugs, medicaments and biological substances	F03.X: Unspecified dementia	J80: Acute respiratory distress syndrome
J44.X: Other chronic obstructive pulmonary disease	F17.X: Nicotine dependence	C34.X: Malignant neoplasm of bronchus and lung	C15.X: Malignant neoplasm of esophagus
I49.X: Other cardiac arrhythmias	R13.X: Aphagia and dysphagia	E66.X: Overweight and obesity	R64: Cachexia
I10.X: Essential (primary) hypertension	R62.X: Lack of expected normal physiological development in childhood and adults	I11.X: Hypertensive heart disease	K65.X: Peritonitis
I99.X: Other and unspecified disorders of circulatory system	K92.X Other diseases of digestive system	M13.X: Other arthritis	C92: Myeloid leukemia

Figure 5: Clustering of Embedding. We perform K-means clustering with K=8 on the embeddings and plot for each of the cluster, the top conditions that are closest to its centroid.

Predicted Cause: I25 Chronic ischemic heart disease			Predicted Cause: J44 Other chronic obstructive pulmonary disease			Predicted Cause: E14 Unspecified diabetes mellitus		
I25	Chronic ischemic heart disease	4.90	J44	Other chronic obstructive pulmonary disease	4.39	J44	Other chronic obstructive pulmonary disease	-1.1
I50	Heart failure	0.68	T82	Complications of cardiac and vascular prosthetic devices, implants and grafts	-1.11	I25	Chronic ischemic heart disease	0.99
T82	Complications of cardiac and vascular prosthetic devices, implants and grafts	-0.64	I25	Chronic ischemic heart disease	-0.91	I50	Heart failure	-0.15
I25	Other chronic obstructive pulmonary disease	-0.34	I50	Heart failure	-0.19	T82	Complications of cardiac and vascular prosthetic devices, implants and grafts	-0.007

Figure 6: Interpretation. We use LIME on our CNN, and obtain the interpretation of the model for top three likely causes of death. For each of the likely cause, we show the contribution of each input medical condition to the final condition, where the absolute values indicate the scale of the contribution, and the sign indicates a positive or negative contribution.

ACKNOWLEDGEMENT

This work was supported in part by grants from the National Center for Advancing Translational Sciences of the National Institutes of Health (NIH) under Award UL1TR000454, National Science Foundation Award NSF1651360, the US Department of Health and Human Services (HHS) Centers for Disease Control and Prevention (CDC) HHSD2002015F62550B, and Microsoft Research and Hewlett Packard. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health nor National Science Foundation. The authors thank Paula Braun and Mark Braunstein for their valuable insights on the project, as well as helpful comments from Ying Sha and Janani Venugopalan.

REFERENCES

[1] Ralph B Agostino. 1998. Tutorial in biostatistics: propensity score methods for bias reduction in the comparison of a treatment to a non-randomized control group. *Stat Med* 17, 19 (1998), 2265–2281.
[2] Kenneth A Bollen and J Scott Long. 1993. *Testing structural equation models*. Vol. 154. Sage.
[3] David Maxwell Chickering. 2002. Optimal structure identification with greedy search. *Journal of machine learning research* 3, Nov (2002), 507–554.
[4] Emerging Risk Factors Collaboration and others. 2011. Diabetes mellitus, fasting glucose, and risk of cause-specific death. *N Engl J Med* 2011, 364 (2011), 829–841.
[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 248–255.
[6] Cícero Nogueira Dos Santos and Maira Gatti. 2014. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts.. In *COLING*. 69–78.
[7] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. 2017. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* 542, 7639 (2017), 115–118.

[8] PyTorch Group. 2017. PyTorch: Tensors and Dynamic neural networks in Python with strong GPU acceleration. http://www.pytorch.org. (2017). Accessed: 2017-04-21.

[9] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation* 18, 7 (2006), 1527–1554.

[10] Andreas Holzinger and Igor Jurisica. 2014. Knowledge discovery and data mining in biomedical informatics: The future is in integrative, interactive machine learning solutions. In *Interactive knowledge discovery and data mining in biomedical informatics*. Springer, 1–18.

[11] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*. 2042–2050.

[12] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

[13] Robert A Israel, Harry M Rosenberg, and Lester R Curtin. 1986. Analytical potential for multiple cause-of-death data. *American journal of epidemiology* 124, 2 (1986), 161–81.

[14] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 2013. 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence* 35, 1 (2013), 221–231.

[15] Hanyu Jiang, Hang Wu, and May D Wang. 2017. A Topic Model View on Causes of Death in the United States, 1999 to 2014. In *Biomedical and Health Informatics (BHI), 2017 IEEE-EMBS International Conference on*. IEEE.

[16] AA Khorana, CW Francis, E Culakova, NM Kuderer, and GH Lyman. 2007. Thromboembolism is a leading cause of death in cancer patients receiving outpatient chemotherapy. *Journal of Thrombosis and Haemostasis* 5, 3 (2007), 632–634.

[17] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[18] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[19] Samantha Kleinberg and George Hripcsak. 2011. A review of causal inference for biomedical informatics. *Journal of biomedical informatics* 44, 6 (2011), 1102–1112.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[21] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent Convolutional Neural Networks for Text Classification.. In *AAAI*, Vol. 333. 2267–2273.

[22] Joy E Lawn, Simon Cousens, Jelka Zupan, Lancet Neonatal Survival Steering Team, and others. 2005. 4 million neonatal deaths: when? Where? Why? *The lancet* 365, 9462 (2005), 891–900.

[23] Yann LeCun, Yoshua Bengio, and others. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* 3361, 10 (1995), 1995.

[24] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3431–3440.

[25] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. 2014. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632* (2014).

[26] Lucie McCoy, Matthew Redelings, Frank Sorvillo, and Paul Simon. 2005. A multiple cause-of-death analysis of asthma mortality in the United States, 1990–2001. *Journal of Asthma* 42, 9 (2005), 757–763.

[27] Alexander Melamed and Frank J Sorvillo. 2009. The burden of sepsis-associated mortality in the United States from 1999 to 2005: an analysis of multiple-cause-of-death data. *Critical Care* 13, 1 (2009), R28.

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

[29] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[30] Pekka K Mölsä, RJ Marttila, and UK Rinne. 1986. Survival and cause of death in Alzheimer's disease and multi-infarct dementia. *Acta Neurologica Scandinavica* 74, 2 (1986), 103–107.

[31] Stephen L Morgan and Christopher Winship. 2014. *Counterfactuals and causal inference*. Cambridge University Press.

[32] Bengt Muthén. 1984. A general structural equation model with dichotomous, ordered categorical, and continuous latent variable indicators. *Psychometrika* 49, 1 (1984), 115–132.

[33] NCHS. 2017. Mortality Data, Vital Statistics NCHS' Multiple Cause of Death Data, 1959 to 2015. (2017). http://www.nber.org/data/vital-statistics-mortality-data-multiple-cause-of-death.html

[34] Judea Pearl. 2009. *Causality*. Cambridge university press.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[36] Matthew D Redelings, Matthew Wise, and Frank Sorvillo. 2007. Using multiple cause-of-death data to investigate associations and causality between conditions listed on the death certificate. *American journal of epidemiology* 166, 1 (2007), 104–108.

[37] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why Should I Trust You?: Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1135–1144.

[38] Paul R Rosenbaum and Donald B Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* (1983), 41–55.

[39] Donald B Rubin. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology* 66, 5 (1974), 688.

[40] Donald B Rubin. 2005. Causal inference using potential outcomes: Design, modeling, decisions. *J. Amer. Statist. Assoc.* 100, 469 (2005), 322–331.

[41] Donald B Rubin. 2006. *Matched sampling for causal effects*. Cambridge University Press.

[42] Donald B Rubin. 2007. The design versus the analysis of observational studies for causal effects: parallels with the design of randomized trials. *Statistics in medicine* 26, 1 (2007), 20–36.

[43] AD Sadovnick, K Eisen, GC Ebers, and DW Paty. 1991. Cause of death in patients attending multiple sclerosis clinics. *Neurology* 41, 8 (1991), 1193–1193.

[44] Peter Spirtes, Clark N Glymour, and Richard Scheines. 2000. *Causation, prediction, and search*. MIT press.

[45] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[46] Elizabeth A Stuart. 2010. Matching methods for causal inference: A review and a look forward. *Statistical science: a review journal of the Institute of Mathematical Statistics* 25, 1 (2010), 1.

[47] Qiuling Suo, Hongfei Xue, Jing Gao, and Aidong Zhang. 2016. Risk Factor Analysis Based on Deep Learning Models. In *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. ACM, 394–403.

[48] Ilya Sutskever, James Martens, and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 1017–1024.

[49] Laurens Van der Maaten. 2014. t-distributed stochastic neighbor embedding (t-SNE). (2014).

[50] Florian Wolf, Tomaso Poggio, and Pawan Sinha. 2006. Human document classification using bags of words. (2006).