This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3069861, IEEE Transactions on Knowledge and Data Engineering

LDA-Reg: Knowledge Driven Regularization using External Corpora

Kai Yang, Zhaojing Luo*, Jinyang Gao, Junfeng Zhao, *Member, IEEE,* Beng Chin Ooi, *Fellow, IEEE,* and Bing Xie, *Member, IEEE*

Abstract—While recent developments of neural network (NN) models have led to a series of record-breaking achievements in many applications, the lack of sufficiently good datasets remains a problem for some applications. For such a problem, we can however exploit a large number of unstructured text corpora as an external knowledge to complement the training data, and most prevailing neural network solutions employ word embedding methods for such purposes. In this paper, we propose LDA-Reg, a novel knowledge driven regularization framework based on Latent Dirichlet Allocation (LDA) as an alternative to the word embedding methods to adaptively utilize abundant external knowledge and to interpret the NN model. For the joint learning of the parameters, we propose EM-SGD, an effective update method which incorporates Expectation Maximization (EM) and Stochastic Gradient Descent (SGD) to update parameters iteratively. Moreover, we also devise a lazy update and sparse update method for the high-dimensional inputs and sparse inputs respectively. We validate the effectiveness of our regularization framework through an extensive experimental study over real world and standard benchmark datasets. The results show that our proposed framework not only achieves significant improvement over state-of-the-art word embedding methods but also learns interpretable and significant topics for various tasks.

Index Terms—Knowledge Driven Regularization, Topic Model, Interpretable Neural Network, Data Analytics, Knowledge Discovery and Data Mining.

1 Introduction

Neural network (NN) models have yielded record-breaking achievements in various areas of application due to their ability to extract meaningful features from the raw data.

The success of NN models is typically associated with large amounts of training data [27]. However, in many real-world tasks, there is a lack of training data. For example, for healthcare applications [22], [33], an Electronic Health Records (EHRs) database usually has only tens of thousands of cases. This always causes the over-fitting problem in NN models and dramatically affects performance.

Interestingly, knowledge associated with different applications is often available in the unstructured data. Take the healthcare application as an example, medical literature from PubMed Central(PMC) which includes more than 5.1 million articles can provide abundant healthcare knowledge. Such abundant external knowledge can be used to complement limited training data and alleviate the issue of over-fitting.

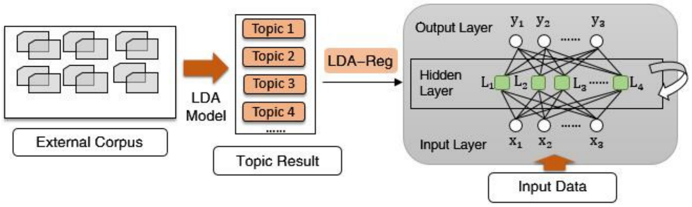


Fig. 1. Overview of LDA-Reg framework.

Techniques based on word embeddings are typically used to incorporate knowledge from external corpora. Their key idea is to learn word vectors to represent each input feature using large-scale external corpora. These pre-trained word embeddings are used to feed into the model in replacement of sparse one-hot representation.

While word embedding methods are proven to be effective, as they capture rich semantic and syntactic relationships learned from external corpora, the external knowledge embedded in the word embeddings is not incorporated into the hidden neurons and less attention has been paid to interpreting the hidden neurons of the NN models which are still regarded as black-box models.

In this paper, we propose LDA-Reg, a novel knowledge driven regularization framework based on Latent Dirichlet Allocation (LDA), as an alternative approach to word embedding methods. This framework incorporates abundant external knowledge into the NN neurons adaptively to the model training process in order to complement the limited training data. In the meantime, it takes advantage of external knowledge to interpret NN neurons for the prediction task.

Our key idea is that documents in the external corpus are composed of words, and the occurrences of a word represent its ”contribution” to a document. This relationship between words and documents is similar to the relationship between

** contact author*

- *K.Yang, J.Zhao and B.Xie are with the School of Electronic Engineering and Computer Science, Peking University, Beijing 100871, China E-mail: {smileyk,zhaojf,xiebing}@pku.edu.cn*
- *Z.Luo, J.Gao and B.C.Ooi are with the Department of Computer Science, Schoool of Computing, National University of Singapore, Singapore 117417, Singapore E-mail: {zhaojing,jinyang.gao,ooibc}@comp.nus.edu.sg*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3069861, IEEE Transactions on Knowledge and Data Engineering

input features and hidden neurons in a NN model where the hidden neurons in the first hidden layer are obtained by weighted summation of input features and each model parameter also represents the ”contribution” of an input feature to a hidden neuron. Based on this observation, we propose LDA-Reg which transfers the hierarchical relationship among the words, topics and documents from external corpora to the relationship among the input features, topics and hidden neurons. By adopting the LDA-Reg, if an input feature is more informative to a hidden neuron according to the external knowledge, less regularization is imposed on the model parameter that connects this input feature with this hidden neuron, leading to a better learned model parameter.

Compared with word embedding methods, LDA-Reg incorporates external knowledge in a deeper manner [9]. To be specific, it incorporates external knowledge directly into neurons of the hidden layers iteratively rather than using external knowledge as input features fed to the input layer. Consequently, in LDA-Reg, external knowledge is an integral part of the training process and the knowledge incorporation process is adaptive to the prediction task.

Further, the benefits of LDA-Reg can be understood from the perspective of representation learning [2]. The hierarchical relationship among documents, topics and words learned from external corpora contain significant statistical information. It can guide different hidden neurons to focus on different topics which further helps discriminate the importance of different input features adaptively. The key idea of LDA-Reg is to impose weaker regularization on the model parameter that connects the hidden neuron with the more important input feature. This adaptive strength of regularization guides the hidden neurons in learning more diversified and discriminative representations by focusing on those more informative input features. In the meantime, it helps hidden neurons to attend to different subsets of input features, which prevent neurons from learning redundant representation and alleviates overfitting [6], [20], [31]. In addition to boosting the model performance, the interpretability of hidden neurons can be easily achieved by inspecting the different topics they focus on.

Figure 1 illustrates the LDA-Reg framework, which is composed of three modules. The first module is to construct NN models for different tasks. The right part of Figure 1 shows a recurrent neural network with one hidden layer. The second module is to pre-train an LDA model on the external corpus. Since each input feature of the NN model built in the first module is a word, we train an LDA on the external corpus for the input features and obtain the topics. The third module is to impose knowledge driven regularization using the topics learned from the external corpus.

Since our proposed regularization is largely determined by the topic mixture parameters , and topic mixture parameters are closely related to the model parameters. Consequently, an efficient method is necessary to update these two sets of parameters appropriately. We proposed to devise an effective update method (EM-SGD) where topic mixture parameters are updated via a lightweight Expectation-Maximization (EM) algorithm, and the model parameters are learned under a common optimization framework, Stochastic Gradient Descent (SGD). In order to reduce the computational costs of EM-SGD, a lazy update and a sparse update method are also proposed for high-dimensional and sparse inputs respectively. Moreover, in contrast to word embedding methods which are difficult to interpret, LDA-Reg is able to interpret the hidden neurons using the learned topics to interpret NN models that are often regarded as black-box models.

Contributions. We make the following contributions:

- We propose a novel and general knowledge driven regularization framework which to our knowledge is the first attempt to use the hierarchical relationships from LDA to regularize the model parameters of the NN model. It is also the first attempt to use topics to interpret hidden neurons.
- We design an efficient update method where topic mixture parameters are updated by lightweight EM algorithm and model parameters are updated via SGD. A lazy update algorithm and a sparse update algorithm are also devised to reduce the computational costs.
- We conduct extensive experiments using both real-world healthcare datasets and sentiment analysis datasets. Results from all the datasets demonstrate that our regularization achieves better performance than L2-norm regularization method [17] and all the baseline word embedding methods such as CBOW [24], skip-gram [24], fasttext [15], GloVe [26], SKG-Topic-Add [32] and SKG-Topic-Concat [32], under their best settings.
- We design LDA-Reg to be flexible and general for its application to any neural network model.

The remainder of the paper is structured as follows. Section 2 reviews related works on topic models and word embedding methods. Section 3 introduces the bayesian interpretation of regularization and Section 4 introduces the LDA-Reg framework. The optimization method is introduced in Section 5. Section 6 introduces LDA-Reg as a flexible framework to incorporate external knowledge. Section 7 reports the experimental results and Section 8 concludes this paper.

2 Related Work

2.1 Topic Models and Word Embedding Methods

Topic models and word embedding methods are two kinds of methods that are widely used to extract knowledge from corpora.

Topic models originated from the field of information retrieval. The main goal for developing the topic models is to summarize large collections of documents efficiently with short and essential descriptions while preserving the statistical relationships [5], [30]. The initial attempt was tf-idf [10]. Although this method could find the words that were discriminative for a document, the inter- and intra-document relationships were not captured. To improve the tf-idf method, LSI [8] method was proposed whose key idea was to employ singular value decomposition on the term-by-document matrix which contained the tf-idf values. LSI could capture more basic linguistic notions such as synonymy and polysemy. Introducing generative probabilistic models into modeling the raw data was a significant progress in this line of research. pLSI, a generative probabilistic model, was proposed in [13] as an alternative of LSI. In this model, each word in a document is innovatively assumed to be generated from a single topic and different words may be generated from different topics. Later, LDA model [4], [12] was proposed to improve pLSI. In LDA, each document is assumed to be generated by a mixture of topics and each topic is represented as a distribution over words. Dirichlet prior was employed as the conjugate prior distribution for better learning of the topic distribution and the word distribution. LDA model is able to capture document-level semantics and discover salient topics of a document. One major advantage of LDA over pLSI is that the topic distribution is treated as random hidden variables rather than a large set of individual parameters that are explicitly linked to the training set. This greatly reduces the number of parameters that need to be learned and makes it easy for LDA to generalize to new unseen documents.

The key idea of word embedding methods is to generate a real-valued vector for each word where rich word relationships are embedded. Traditional word embedding methods adopt the neural network language model (NNLM) [3], which is very computationally expensive. In order to reduce computational costs, many simple models have been proposed [15], [24]. CBOW and skip-gram are two simple representative models for learning word embeddings. They extend NNLM, but learning the word embeddings using a simple model. The difference between CBOW and skip-gram is that the CBOW predicts the current word based on the context, while skip-gram predicts surrounding words given the current word. Although the word embeddings learned by these two methods capture rich semantic and syntactic relationships, they put less emphasis on global information. Another line of research which employs matrix factorization for low-dimensional word embeddings [18] focuses more on global information. These methods consider global statistics, but the local information is needed to be incorporated for better performance. Results in [26] show these word embedding methods generate a sub-optimal structure. GloVe [26] improves the two kinds of methods mentioned above by training on a co-occurrence matrix so as to combine the advantages of the above two kinds of methods.

LDA and word embedding methods capture semantics differently. LDA summarizes documents using topics and provides an explicit representation of a document. It describes the hierarchical relationships among words, topics and documents. In contrast, word embeddings describe word-word relationships and learn similar word embeddings for similar words. In terms of representation, LDA provides probability distributions that describe the statistical relationship while word embeddings embed word-level semantics in low-dimensional dense word vectors [28]. When embedding methods are used to incorporate external knowledge into the NN models, the pre-trained word vectors are fed as the input to the model. This makes it hard to incorporate knowledge adaptively to the specific prediction task. On the contrary, in the LDA-Reg, the hierarchical relationship captured by LDA from an external corpus is incorporated into the NN model in a deeper way, i.e., into the hidden neurons and hidden layers of the NN models, so that the external knowledge is made an integral part of the learning process and the incorporation of the external knowledge is adaptive to the prediction task.

2.2 Combining Topic Models with Word Embedding Methods

In order to better take advantage of both global and local information, there are related works which combine topic models and word embedding methods. EETM [34] is an embedding enhanced topic model where topic information is transmitted into the topic embeddings by leveraging the word embeddings so that these two kinds of embeddings share high-level semantic information sufficiently for better topic-level representation. GaussianLDA [7] is another work which enhances the topic model using word embeddings. Its key idea is to represent the words using word embeddings in the LDA model so that words that are semantically related in the embedding space can be grouped into the same topic and the generated topics are more semantically coherent.

Different from EETM [34] and GaussianLDA [7] which are topic models, the neural language model proposed in [32] is a topic-based word embedding method. Its key idea is to capture the topic-based word relationship with LDA and then incorporate it into the word embedding learning so as to better utilize statistical information about the corpus for generating word embeddings. In practice, the learned topic-based word embeddings are added or concatenated with other word embeddings, e.g.,skip-gram, and are denoted as SKG-Topic-Add [32] and SKG-Topic-Concat [32] respectively.

For EETM [34] and GaussianLDA [7], both are topic models and hence it is convenient for our proposed LDA-Reg framework to work with them by replacing the generation probability of the words with their customized ones. Details are explained in Section 6. While in [32], the proposed method is an enhanced version of the word embedding method. Consequently, when incorporating external knowledge into NNs using this method, the learned word embeddings are fed into the input layer and the incorporation of external knowledge is not adaptive to the prediction task.

3 Bayesian Interpretation of Regularization

In order to take advantage of the external knowledge, we propose to incorporate the external knowledge into the regularization term of the model. Before explaining the idea, we need first to understand the regularization term from the Bayesian perspective.

From the perspective of Bayes' theorem, the regularization is regarded as the prior distribution of the model parameters $\boldsymbol{v}$. In Bayes' rule, the posterior probability of model parameters $\boldsymbol{v}$ is $p(\boldsymbol{v}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{v})p(\boldsymbol{v})}{p(\mathcal{D})}$. Here $\mathcal{D}$ denotes the observed data, $\boldsymbol{v}$ denotes the model parameters, $p(\mathcal{D}|\boldsymbol{v})$ denotes the likelihood function and $p(\mathcal{D})$ corresponds to a constant. Model parameters $\boldsymbol{v}$ are usually optimized using maximum a posterior (MAP) estimation [16], which is written as $\boldsymbol{v}_{MAP} = \underset{\boldsymbol{v}}{\text{argmin}}\ (-\log p(\mathcal{D}|\boldsymbol{v}) - \log p(\boldsymbol{v}))$. The term $\log p(\boldsymbol{v})$ is the regularization term which is log of the prior distribution for model parameters. Many related works [16], [19], [21] focus on modeling model parameters prior distribution. Typically, if Laplacian distribution and Gaussian distribution are used as $p(\boldsymbol{v})$, the L1-norm and L2-norm regularization can be derived respectively. In our work, we assume $p(\boldsymbol{v})$ is related to the LDA model learned from the external corpus and derive a regularization function based on LDA.

4 LDA-Reg Framework

Our main idea is that documents in the external corpus are composed of words, and the occurrences of a word represent its ”contribution” to a document. This relationship between words and documents is similar to the relationship between input features and hidden neurons in a NN model. The hidden neurons in the first hidden layer are obtained by weighted summation of input features and each model parameter also represents the ”contribution” of an input feature to a hidden neuron.

Based on this observation, LDA-Reg is designed to transfer the hierarchical relationship among the words, topics and documents from external corpora to the relationship among the input features, topics and hidden neurons. If an input feature is more informative to a hidden neuron according to the external knowledge, less regularization is imposed on the model parameter that connects this input feature with this hidden neuron.

To exploit the hierarchical knowledge from the external corpus, the word mixtures of different topics, denoted by Φ, is first learned from the external corpus, and then it is shared and utilized in the NN model for the prediction task. Specifically, in the NN model, the generation probabilities for each input feature and each hidden neuron are designed according to the LDA model [4], [12] by taking account of both Φ and model parameters. To incorporate the external knowledge is essentially to maximize the generation probabilities of all the hidden neurons and input features.

To achieve the above-mentioned idea, the LDA-Reg framework consists of three modules: building the NN model, training LDA on the external corpus to obtain the word mixtures Φ and imposing knowledge driven regularization. In the remainder of this section, we will introduce these three modules in detail.

4.1 Neural Network Model and LDA on External Corpora

In the LDA-Reg framework, we denote the neurons of the first hidden layer as knowledge-related neurons and we denote the model parameters that connect the first hidden layer with the input features as knowledge-related model parameters, $\boldsymbol{v}^r$. Our knowledge driven regularization is applied on $\boldsymbol{v}^r$, and for the other model parameters $\boldsymbol{v}^o$, L2-norm regularization is used as the regularization method. Thus in the following sections, we mainly focus on the knowledge-related model parameters $\boldsymbol{v}^r$. For the NN model, the Negative log-likelihood is set as the loss function.

$$L(\boldsymbol{v}^r, \boldsymbol{v}^o) = -\log p(\mathcal{D}|\boldsymbol{v}^r, \boldsymbol{v}^o) \tag{1}$$

, which corresponds to the first term in the MAP estimation introduced in Section 3.

In terms of training the LDA model [4], [12] on the external corpus, the words of external corpora that can not be recognized as input features of the NN model are deleted firstly. Next, we train an LDA model on the processed external corpora to make sure that the learned topics are the mixtures of input features from the NN model. After the LDA model is trained, the word mixtures of different topics, denoted by Φ, is obtained and then used for knowledge driven regularization explained in Section 4.2.

4.2 Knowledge Driven Regularization

Knowledge driven regularization is designed to incorporate the hierarchical relationship among words, topics and documents into the NN model effectively by utilizing the word mixtures Φ learned from the external corpus. To achieve this, we need to first define the generation probabilities for the input features and hidden neurons by using the Φ.

In LDA-Reg, each input feature is regarded as a word. Take the Sentence Polarity dataset from Section 7.1 as an example. On example input is ”It is a fantastic movie.” where each word is an input feature. The generation probability of each input feature/word j of the i-th knowledge-related neuron is designed according to the LDA model [4], [12]:

$$\begin{aligned} p(w_{i,j}|\overrightarrow{\theta}_i, \Phi) &= \sum_{k=1}^{K}\{p(w_{i,j}|\overrightarrow{\varphi}_k)p(z_{i,j}=k|\overrightarrow{\theta}_i)\} \\ &= \sum_{k=1}^{K}\{\varphi_{k,w_{i,j}}\theta_{i,k}\} \end{aligned} \tag{2}$$

where $w_{i,j}$ means the j-th input feature, i.e., one of the words in the input sentence, for the i-th knowledge-related neuron. Accordingly, $\overrightarrow{\theta}_i$ is the topic mixture for the knowledge-related neuron i and $\overrightarrow{\theta}_i$ needs to be learned during the NN model training process. Φ is learned from the external corpus, K is the number of topics, $\overrightarrow{\varphi_k}$ is the word mixture for topic k, $z_{i,j}$ is the topic indicator.

In the standard LDA model, the occurrences of a word represent its ”contribution” to a document. In comparison, in LDA-Reg, the absolute value of the model parameter is regarded as the ”contribution” of an input feature to a knowledge-related neuron. Specifically, for knowledge-related neuron i, when the contribution of the j-th input feature $|v_{i,j}^r|$ is considered, the generation probability of this hidden neuron i is written as:

$$p(\overrightarrow{w_i}, \overrightarrow{v_i}^r|\overrightarrow{\theta_i}, \Phi, \lambda) = \prod_{j=1}^{J}\{p(w_{i,j}|\overrightarrow{\theta_i}, \Phi)^{\lambda|v_{i,j}^r|}\} \tag{3}$$

where J is the number of input features, λ is a hyperparameter that controls the contribution strength of the input feature.

Lastly, Dirichlet distribution, which is the conjugate prior for topic mixture parameters $\overrightarrow{\theta_i}$, is imposed to control how uniform the generated $\overrightarrow{\theta_i}$ are. The joint distribution for $\overrightarrow{w}_i$, $\overrightarrow{v}_i^r$, $\overrightarrow{\theta}_i$ of knowledge-related neuron i can thus be written as:

$$p(\overrightarrow{w}_i, \overrightarrow{v}_i^r, \overrightarrow{\theta}_i|\Phi, \overrightarrow{\alpha}, \lambda) = p(\overrightarrow{w}_i, \overrightarrow{v}_i^r|\overrightarrow{\theta}_i, \Phi, \lambda)p(\overrightarrow{\theta}_i|\overrightarrow{\alpha}) \tag{4}$$

When considering all the knowledge-related neurons, the joint distribution for $\overrightarrow{w}_i$, $\overrightarrow{v}_i^r$, $\overrightarrow{\theta}_i$ of all knowledge-related neurons can be written as: $\prod_{i=1}^{I}\{p(\overrightarrow{w}_i, \overrightarrow{v}_i^r, \overrightarrow{\theta}_i|\Phi, \overrightarrow{\alpha}, \lambda)\}$, where I is the number of knowledge-related neurons in the NN model.

4.3 Overall Loss Function

Given a real-world task, our goal is to incorporate external knowledge into regularization in order to complement the training data as well as interpret the neural network model. According to MAP estimation introduced in Section 3, the overall loss function G is defined as follows:

$$G = -\log p(\mathcal{D}|\boldsymbol{v}^r, \boldsymbol{v}^o) - \sum_{i=1}^{I} \log\{p(\overrightarrow{w}_i, \overrightarrow{v}_i^r, \overrightarrow{\theta}_i|\Phi, \overrightarrow{\alpha}, \lambda)\} \tag{5}$$

where the first term corresponds to the loss function of the NN model and the second term corresponds to the knowledge driven regularization related to external knowledge.

5 Optimization (EM-SGD)

In LDA-Reg, two sets of correlated parameters need to be updated, i.e., knowledge-related model parameters $\boldsymbol{v}^r$ and topic mixture parameters $\overrightarrow{\theta}_i$ of each knowledge-related neuron i. We propose to update both of them jointly from the joint distribution defined in Equation 5. If we fix topic mixture parameters $\overrightarrow{\theta}_i$, Equation 5 actually reduces to MAP with the model parameter prior defined by the specific choice of $\overrightarrow{\theta}_i$. From this perspective, the optimization problem defines a series of MAP inference problems. This suggests we can devise a natural iterative algorithm where $\boldsymbol{v}^r$ and $\overrightarrow{\theta}_i$ are optimized alternatively until convergence. Specifically, at a high-level, the update method consists of SGD and the EM algorithm. Concretely, for knowledge-related model parameters, SGD is used as the update method. For topic mixture parameters, a lightweight EM algorithm is designed.

Figure 2 shows how SGD interacts with EM in our update method. After both kinds of parameters are initialized, topic mixture parameters of knowledge-related neurons are calculated. The regularization is then calculated and affects the knowledge-related model parameters through SGD. After the knowledge-related model parameters are updated via an SGD step, one step of EM is employed to update the topic mixture parameters based on the updated knowledge-related model parameters. Subsequently, a new regularization is calculated for the knowledge-related model parameters. This process iterates until the algorithm converges. Section 5.1 introduces the SGD method for updating knowledge-related model parameters and Section 5.2 introduces the EM algorithm for updating topic mixture parameters.

5.1 Stochastic Gradient Descent Part

When topic mixture $\overrightarrow{\theta}_i$ of knowledge-related neuron i is fixed, gradient descent method is employed to update the knowledge-related model parameters $\boldsymbol{v}^r$. According to Equation 5, the gradient for $v_{i,j}^r$ with respect to the overall loss function G is

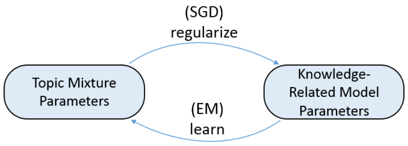


Fig. 2. Overview of EM-SGD.

$$\frac{\partial G}{\partial v_{i,j}^r} = -\frac{\partial \log p(\mathcal{D}|\boldsymbol{v}^r, \boldsymbol{v}^o)}{\partial v_{i,j}^r} - \lambda sign(v_{i,j}^r)\log(\sum_{k=1}^{K}\varphi_{k,w_{i,j}}\theta_{i,k}) \tag{6}$$

where $\varphi_{k,w_{i,j}}$ is the word mixture value for the j-th input feature $w_{i,j}$ of the i-th knowledge-related neuron in topic k and $\theta_{i,k}$ is the topic mixture value of topic k in the i-th knowledge-related neuron.

In Equation 6, the first part is the gradients with respect to the negative log-likelihood and the second part is the gradients with respect to the knowledge driven regularization. This equation shows that LDA-Reg imposes a regularization similar to L1-norm regularization. The regularization strength on $v_{i,j}^r$ is related to $\sum_{k=1}^{K}\varphi_{k,w_{i,j}}\theta_{i,k}$, which is the generation probability for the j-th input feature $w_{i,j}$ of the i-th knowledge-related neuron. In other words, according to the external knowledge, if the generation probability of $w_{i,j}$ is high, it means the j-th input feature is very informative to the i-th knowledge-related neuron. According to equation 6, less regularization is imposed on the knowledge-related model parameter $v_{i,j}^r$, which connects the j-th input feature with the i-th knowledge-related neuron.

This adaptive regularization guides the hidden neurons to attend to more informative neurons so that different hidden neurons are able to learn more diversified and discriminative representations, which prevents neurons from learning redundant representation and alleviates overfitting.

5.2 EM Algorithm Part

The update of the topic mixture $\overrightarrow{\theta}_i$ for each knowledge-related neuron i includes the E-step and the M-step. The update equations of the EM algorithm is introduced in the following subsections.

E-Step. We calculate a function called the responsibility in the E-step. For a knowledge-related neuron i, given topic mixtures $\boldsymbol{\Theta}$ and word mixtures Φ, the responsibility of topic q for the j-th input feature $w_{i,j}$ is defined as follows:

$$r_{i,q}(w_{i,j}) = \frac{\theta_{i,q}\varphi_{q,w_{i,j}}}{\sum_{k=1}^{K}\{\theta_{i,k}\varphi_{k,w_{i,j}}\}} \tag{7}$$

This function can be interpreted as calculating the conditional probability that in knowledge-related neuron i, a particular j-th input feature $w_{i,j}$ is generated by a particular topic q.

M-Step. After the responsibilities are updated in E-step, the topic mixture parameters $\overrightarrow{\theta}_i$ of knowledge-related neuron i are updated using the current responsibilities. This is the M-Step. This section introduces the update formula for topic mixture parameters.

The gradient of $\theta_{i,q}$ with respect to G is as follows:

$$\frac{\partial G}{\partial \theta_{i,q}} = -\frac{\alpha_q - 1}{\theta_{i,q}} - \sum_{j=1}^{J}\{\frac{r_{i,q}(w_{i,j})}{\theta_{i,q}}\lambda|v_{i,j}^r|\} \tag{8}$$

where $\theta_{i,q}$ is the q-th dimension of $\overrightarrow{\theta}_i$.

Equation 8 shows that the first term is controlled by hyperparameter $\overrightarrow{\alpha}$ and the second term is related to responsibility function.

Given fixed responsibility function value, the minimizer for $\theta_{i,q}$ can be derived by setting Equation 8 to zero. However, according to LDA model, $\overrightarrow{\theta}_i$ is the topic mixture of the knowledge-driven neuron i and it is the parameter of the multinomial distrbution. Consequently, the condition $\sum_{k=1}^{K}\theta_{i,k} = 1$ must be satisfied. We thus propose to utilize the Lagrange multiplier method to address this issue.

Since the $\overrightarrow{\theta}_i$ of each knowledge-related neuron i is calculated independently, we introduce μ_i as the Lagrange multiplier for $\overrightarrow{\theta}_i$. The Lagrangian of the loss function is

$$L = G + \sum_{i=1}^{I}\{\mu_i(\sum_{k=1}^{K}\theta_{i,k} - 1)\} \tag{9}$$

After setting the gradient of $\theta_{i,q}$ and μ_i with respect to L to zero, we obtain the update formula for topic mixture parameters:

$$\theta_{i,q} = \frac{(\alpha_q - 1) + \sum_{j=1}^{J}\{r_{i,q}(w_{i,j})\lambda|v_{i,j}^r|\}}{\sum_{k=1}^{K}\{(\alpha_k - 1) + \sum_{j=1}^{J}\{r_{i,k}(w_{i,j})\lambda|v_{i,j}^r|\}\}} \tag{10}$$

Equation 10 shows that the topic mixture parameters are related to two factors. The first is the Dirichlet hyperparameter which has the smoothing effects and the second is the weighted sum of the absolute values of the model parameters where the weights are responsibilities. Furthermore, topic mixture parameters can also be interpreted as containing both external knowledge via responsibility and the internal model information via the model parameters.

With $\theta_{i,q}$, LDA-Reg is able to adaptively differentiate different knowledge-related neurons by guiding them to attend to different topics.

5.3 Practical Design Considerations

In order to apply LDA-Reg to real-world large-scale datasets, we need to reduce the computational costs of LDA-Reg. Empirically, one can make two observations. First, the update of responsibilities and topic mixture parameters, which correspond to E-step and M-step respectively, are time-consuming because they involve large matrix operations. Second, the inputs are one-hot representations, which are extremely sparse. We can exploit these findings to avoid the naive approach of updating LDA-Reg. Therefore, we devise a lazy update and a sparse update method for high-dimensional inputs and sparse inputs respectively.

5.3.1 *Lazy Update*

The update of responsibilities and topic mixture parameters involve large matrix operations and thus are very time-consuming. Fortunately, both responsibilities and topic mixture parameters do not change too much after the first few epochs. A computationally efficient approximation of these two parameters is to update them every few iterations instead of every iteration, which is the key idea of the lazy update.

Algorithm 1 shows the lazy update algorithm (LEM-SGD). $\overrightarrow{\alpha}$ is the Dirichlet hyperparameter and Θ is the topic mixture parameters for all knowledge-related neurons. It is initialized such that each topic of each knowledge-related neuron has the same probability. Φ is learned by the LDA model on external corpora. lr is the learning rate for SGD, E is the number of the first few epochs when the lazy update is not employed (For simplicity purposes, we call E as first epochs hyperparameter) and B is the number of mini-batches in the training dataset. I_{EM} is the update interval for responsibilities and topic mixture parameters $\mathbf{\Theta}$. In this algorithm, the iteration counter it is first initialized to 0. After that, the gradients with respect to negative log-likelihood are calculated. Subsequently, the algorithm computes responsibilities and topic mixture parameters $\mathbf{\Theta}$ via one EM step and updates knowledge-related model parameters via one SGD step. Note that the EM step is carried out only every I_{EM} iterations instead of each iteration.

5.3.2 *Time Analysis for Lazy Update*

In this section, We provide theoretical time analysis for the lazy update by analyzing the effect of update interval I_{EM} as well as the first epochs hyperparameter E. From Algorithm 1, we note that the overall computation consists of four steps, namely gradients computation concerning the negative log-likelihood (line 3), E step, M step and SGD step. We use t_{nll} to denote the time of the gradients computation concerning the negative log-likelihood for each iteration. R_L and R_N respectively indicate the ratio of computation time for E step, M step and SGD step when employing and not employing lazy update to t_{nll}.

Further, we use P and B to denote the total number of epochs and the total number of mini-batches every epoch. The total time T for different I_{EM} and E is calculated as follows:

$$\begin{aligned} T = {} & t_{nll} \times B \times E \times (1 + R_N) \\ & + t_{nll} \times B \times (P - E) \times \\ & (1 + (1 - 1/I_{EM}) \times R_L + R_N/I_{EM}) \end{aligned} \tag{11}$$

Equation (11) consists of two terms, the first term calculates the time of the first E epochs without the lazy update and the second term represents the time of the remaining epochs with the lazy update. After reformulating Equation (11), we obtain:

$$\begin{aligned} T = {} & t_{nll} \times B \times \\ & [P \times (1 + R_L) + (E + \frac{(P - E)}{I_{EM}}) \times (R_N - R_L)] \end{aligned} \tag{12}$$

From this Equation, we can see E and I_{EM} affect the computation time through term $E + \frac{(P-E)}{I_{EM}}$. Given $P \geq E$, E has a positive correlation with the total time T and I_{EM} has a negative correlation with the total time T. Consequently, in practice, it is recommended that E is set to a relatively small value and I_{EM} to a relatively large value.

Algorithm 1 Update for LDA-Reg with LEM-SGD

```
Input: $\boldsymbol{v}^r$, $\boldsymbol{v}^o$, $\overrightarrow{\alpha}$, $\Theta$, $\Phi$, $lr$, $E$, $B$, $I_{EM}$
Output: $\boldsymbol{v}^r$, $\Theta$
initialize: $it \leftarrow 0, epoch\_it \leftarrow 0$
while not converged do
  Compute $\frac{\partial - \log p(\mathcal{D}|\boldsymbol{v}^r, \boldsymbol{v}^o)}{\partial \boldsymbol{v}^r}$
/* E-step */
  if $epoch\_it < E$ or $it \ mod \ I_{EM} = 0$ then
    Compute responsibilities based on Equation (7)
  end if
  Compute $\frac{\partial G}{\partial \boldsymbol{v}^r}$ based on Equation (6)
/* M-step */
  if $epoch\_it < E$ or $it \ mod \ I_{EM} = 0$ then
    Compute $\boldsymbol{\Theta}$ for all knowledge-related neurons based on Equation (10)
  end if
/* SGD-step */
  $\boldsymbol{v}^{r(it+1)} \leftarrow \boldsymbol{v}^{r(it)} - lr \frac{\partial G}{\partial \boldsymbol{v}^r}$
  $it \leftarrow it + 1$
  if $it \ mod \ B = 0$ then
    $epoch\_it \leftarrow epoch\_it + 1$
  end if
end while
```

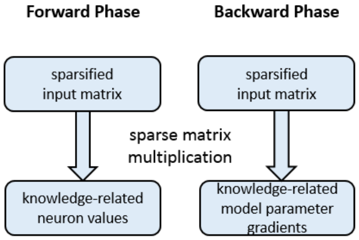

Forward Phase
Backward Phase
sparsified input matrix
sparsified input matrix
sparse matrix multiplication
knowledge-related neuron values
knowledge-related model parameter gradients

Fig. 3. Sparse update.

5.3.3 Sparse Update

For neural network models with knowledge driven regularization, the inputs are one-hot representations, which are extremely sparse. Introducing sparse matrix operations into the forward and backward phases of the model is able to largely reduce the overall computational costs.

The detailed process of sparse update is shown in Figure 3. In the forward phase, the input matrix is first transformed to the sparse format [1]; subsequently, this sparsified input matrix is multiplied with the knowledge-related model parameters in order to calculate the knowledge-related neuron values. In the backward phase, the sparsified input matrix is multiplied with the gradients of the knowledge-related neurons so as to calculate the gradients of the knowledge-related model parameters. The overall computational costs can be reduced largely by performing sparse multiplications in both forward and backward phases.

5.3.4 Time Analysis for Sparse Update

The total time of NN model training can be divided into two parts, i.e., the total time for the first layer and the other layers. Although the sparse update is implemented for the first layer, analyzing the time speedup brought about by sparse update needs to take other layers into consideration. We thus formalize the time speedup S due to the sparse update as $S = \frac{1}{(1-p)+\frac{p}{k}}$. Here $p(0 \leq p \leq 1)$ is the portion of the training time of the first layer out of all the layers, and k is the speedup of the first hidden layer brought about by the sparse update. From this equation, we note that both increased k and increased p lead to an increased speedup S. For the sparse update, it decreases the training time for the first layer. Consequently, the k value is larger than 1, leading to the speedup S being larger than 1, which saves the total training time.

6 A Flexible Framework to Incorporate External Knowledge

Since our LDA-Reg framework retains the hierarchical structure among documents, topic and word, in addition to standard LDA, LDA-Reg can also work with other topic models [7], [34] which may further improve the model performance. To be specific, to integrate other topic models, the only change to LDA-Reg is to replace the generation probability of the input word defined in Equation 2 with their customized ones.

7 Experiments

In this section, we evaluate the effectiveness of our proposed LDA-Reg framework in terms of predictive performance and interpretation on two applications, namely, disease prediction and sentiment analysis. By choosing these two applications, we aim to show the effectiveness of LDA-Reg in the applications that require abundant external knowledge and the applications that are typical Natural Language Processing (NLP) tasks. The baseline methods include L2-norm regularization (L2-Reg) method, which does not incorporate external knowledge, and state-of-the-art word embedding methods, including CBOW [24], skip-gram [24], fasttext [15], GloVe [26], SKG-Topic-Add [32] and SKG-Topic-Concat [32]. The last two word embedding methods combine the LDA model.

7.1 Datasets

MIMIC-III Dataset: MIMIC-III [14] is a public benchmark dataset that includes various types of medical events generated by patients, such as diagnoses (e.g., ”Dissection of Coronary Artery”, ”Coronary Artery Anomaly” and ”Congestive Heart Failure”), lab tests (e.g., ”Phosphate”, ”Alkaline Phosphatase” and ”Rocaltrol”), medications (e.g., ”Calcium Carbonate”, ”Phospha 250 Neutral” and ”Sodium CITRATE 4”), etc. In this dataset, we study an 80-class classification problem that predicts the diseases of a patient given the medical history of 90 days. This dataset consists of 19343 samples, each of which has 4351 features.

We transform the irregular medical time series data into a regular one through resampling the data into 9 disjoint windows, taking the counts of the medical events within each 10-day window.

Sentence Polarity Dataset: Sentence Polarity Dataset v1.0 [25] is a public benchmark dataset for sentiment analysis. It includes 5331 positive and 5331 negative processed movie reviews. Each sentence is labeled with its overall sentiment polarity. One example sentence with a positive label is ”It is a fantastic movie.”, where each word is regarded as an input feature. In this dataset, we perform sentiment analysis, which is a binary classification task. There are in total 10662 samples, each of which has 5229 features.

[1] https://docs.scipy.org/doc/scipy/reference/sparse.html

7.2 External Corpora

PubMed Dataset: We use PubMed[2], a free search engine accessing numerous online medical literature as our external corpus for disease prediction task using MIMIC-III dataset. For PubMed medical literature, we focus on the publications that contain the medical features in the MIMIC-III dataset. There are in total 53166 qualified publications.

Large Movie Review Dataset (LMRD): We use Large Movie Review Dataset [23] as our external corpus for the sentiment analysis task using Sentence Polarity dataset. For this corpus, we focus on the movie review documents that contain the words in the Sentence Polarity dataset. There are in total of 50000 qualified documents.

Wikipedia Dataset (WIKI): For the sentiment analysis task, we also use Wikipedia[3] as another external corpus. By using this corpus, we aim to show the performance of our LDA-Reg on the general domain knowledge corpus. For this corpus, we focus on the articles that contain the words in the Sentence Polarity dataset. There are in total of 100000 qualified documents.

7.3 Data Preprocessing

7.3.1 Training LDA and Word Embeddings on External Corpus

Before training the LDA model and word embeddings, the words of external corpora that can not be recognized as input features of our NN model are deleted.

We train an LDA over each processed external corpus separately and the learned topics are the mixtures of input features. These LDAs are trained using the gensim package [4].

For word embeddings, the gensim package is used to learn CBOW, skip-gram, SKG-Topic-Add [32] and SKG-Topic-Concat [32] word embeddings, the GloVe package [5] is used to learn GloVe word embeddings, the fasttext package [6] is used to learn fasttext word embeddings.

7.3.2 Organizing Inputs

For word embedding methods, the learned embeddings of different features are averaged before they are input into the MLP model. For the LSTM model, the learned embeddings of different features at the same timestamp are also averaged as the final inputs.

While for LDA-Reg and L2-Reg, the input data to MLP/LSTM model is preprocessed using the bag-of-words method. To be specific, we use the medical event count vectors as inputs on the MIMIC-III dataset and one-hot encodings of words as inputs on the Sentence Polarity dataset.

7.3.3 Splitting Datasets

We divide the whole dataset into a random 6.4-1.6-2 training-validation-test split. Then we use the training dataset for training our MLP/LSTM model, validation dataset for determining the hyperparameters. The test dataset is used for testing the performance of the model. Each experiment is run for five times and the average results and standard deviation are reported.

7.4 Experimental Settings

MLP and LSTM Models: Two kinds of neural network models are employed in our experiment. The first kind is MLP and the second kind is LSTM. By conducting experiments on these two models, we aim to show the effectiveness of LDA-Reg on both simple neural network models and complex neural network models that are designed for time series inputs. In terms of the number of hidden layers, we experiment with both one hidden layer and two hidden layers.

We implement our MLP and LSTM models in Pytorch[7]. For MLP, the hidden size of the models is 128. Both the hidden layer and the output layer set the sigmoid function as the activation function. For LSTM models, the hidden size of the models is 128. They include cell state, input gate, forget gate and output gate. The input gate, forget gate and output gate take sequential data as well as the last hidden state as input and set sigmoid function as active function. While cell state and hidden state set tanh function as active function. The activation function for the output layer is sigmoid. The input sequence lengths for the MIMIC-III dataset and the Sentence Polarity dataset are 9 and 25 respectively.

For word embedding methods and L2-Reg, weight decay is employed on all model parameters. For LDA-Reg, knowledge driven regularization is applied on the model parameters connecting the first hidden layer and input features. For the other model parameters, weight decay is employed.

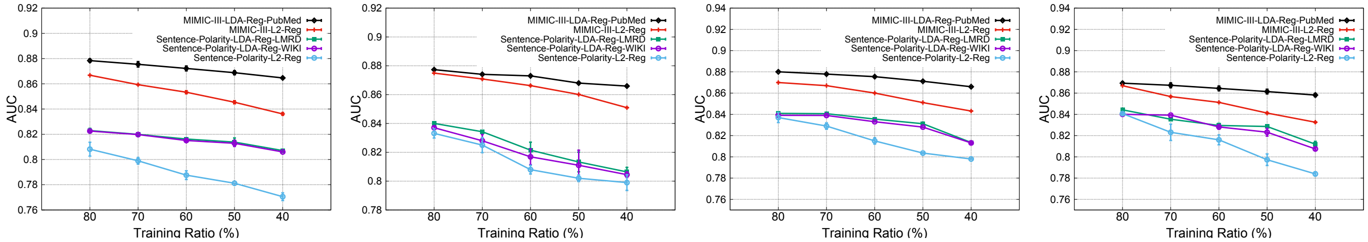
Hyperparameters: Hyperparameters of both LDA-Reg and baseline methods are tuned using the validation dataset.

In terms of the MIMIC-III dataset, for both MLP and LSTM models, the optimizer is the Adam gradient method with momentum as 0.9. The learning rate and weight decay are set to 0.001 and 0.0001 respectively. The batch size is 128 and the number of training epochs is 600. For LDA-Reg, the topic number is 50. λ and α are both set to 1. For L2-Reg, the regularization strength is set to 0.001. For word embedding methods, the embedding size is set to 500.

In terms of the Sentence Polarity dataset, for both MLP and LSTM models, the optimizer is the Adam gradient method with momentum as 0.9. The learning rate and weight decay are set to 0.001 and 0.0001 respectively. The batch size is 128 and the number of training epochs is 600. For LDA-Reg, the topic number is 200. λ and α are set to 0.001 and 1 respectively. For L2-Reg, the regularization strength is set to 0.001. For word embedding methods, the embedding size is set to 500.

Evaluation Metric: To evaluate the proposed LDA-Reg, we use the metric, Area Under the receiver operating characteristic Curve (AUC) to measure the classification performance. A receiver operating characteristic curve is a graph showing the performance of a classification model at all classification thresholds. The AUC is then calculated as the area under the

[2] https://www.ncbi.nlm.nih.gov/pubmed/
[3] https://dumps.wikimedia.org/enwiki/latest/
[4] https://radimrehurek.com/gensim/
[5] https://github.com/stanfordnlp/GloVe
[6] https://pypi.python.org/pypi/fasttext
[7] http://pytorch.org/

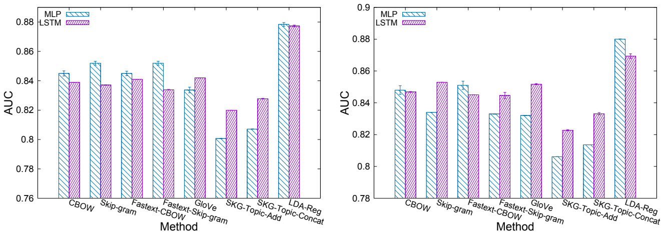


(a) One layer MLP (b) One layer LSTM (c) Two layer MLP (d) Two layer LSTM

Fig. 4. AUC for different training ratios.

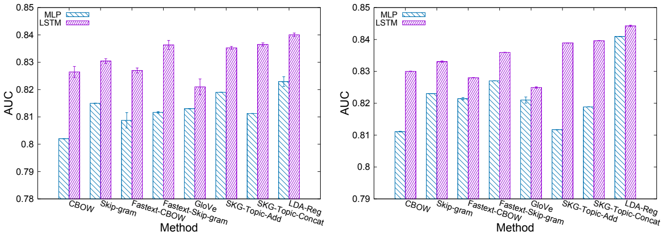
curve [11] and it measures the model's ability to distinguish between different classes. Larger AUC values indicate better performance of the model. For the MIMIC-III dataset, since the task is an 80-class classification problem, we compute the AUC across all classes as the evaluation metric.

Environment: Experiments are run on a server equipped with the Intel Xeon E5-2620 v4 CPU and four Titan Xp GPU cards.



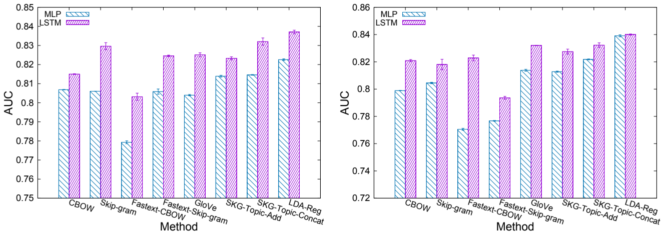
(a) One layer models (b) Two layer models

Fig. 5. AUC for MIMIC-III dataset.



(a) One layer models (b) Two layer models

Fig. 6. AUC for Sentence Polarity dataset using LMRD corpus.



(a) One layer models (b) Two layer models

Fig. 7. AUC for Sentence Polarity dataset using WIKI corpus.

7.5 Predictive Performance on Two Real-World Datasets

In this section, we compare our LDA-Reg with L2-Reg as well as seven state-of-the-art word embedding methods on two real-world datasets.

7.5.1 *Comparison with L2-Reg*

In order to show the benefits of incorporating external knowledge, we compare the performance between NN models with LDA-Reg and NN models with L2-Reg which does not incorporate external knowledge. As mentioned in Section 7.3.2, the inputs of LDA-Reg and L2-Reg are both sparse one-hot representation. The only difference between these two methods is that L2-Reg employs L2-norm regularization (weight decay) on all the model parameters. In contrast, LDA-Reg employs knowledge driven regularization on the knowledge-related model parameters and employs L2-norm regularization (weight decay) on the other model parameters. Specifically, we compare these two regularization methods by varying the amount of training data while keeping test data unchanged. Since L2-Reg does not incorporate external knowledge, we want to investigate the effectiveness of incorporating external knowledge when the training data is not sufficient by varying the amount of training data. By default, 80% of the data is used for training while 20% is used for testing. In this experiment, we vary the training ratio from 80% to 40% and evaluate the trained models on the same test data.

Figure 4 shows the AUC of LDA-Reg and L2-Reg for different training ratios in two datasets for both one layer and two layer models. In both datasets, MLP and LSTM models with LDA-Reg dominate those with L2-Reg. Also, the AUC of the models with L2-Reg decreases faster than that of the models with LDA-Reg as the training ratio decreases. The improvement of LDA-Reg over L2-Reg is mainly attributed to the fact that LDA-Reg utilizes the semantic information contained in external knowledge. Specifically, the external knowledge, which contains hierarchical information of topics and words learned from a large number of external corpora, guides the model to differentiate different hidden neurons by enabling them to focus on different topics to discriminate the importance of each input feature adaptively. Weaker regularization is then imposed on the model parameter that connects the more important input feature. This adaptive customized regularization helps the hidden neurons learn more diversified and discriminative representations so as to alleviate overfitting [6], [20], [31]. In comparison, L2-Reg imposes the same strength of regularization on all the model parameters without differentiating different input features.

In terms of comparison between LMRD and WIKI, we observe that LDA-Reg with LMRD as external corpus performs slightly better than that with WIKI because LMRD is more related to the movie review application. Nevertheless, LDA-Reg with WIKI as external corpus still achieves better results than L2 Reg, which confirms the effectiveness of our knowledge driven regularization.

The slighter decrease in AUC of LDA-Reg with respect to the training ratio decrease demonstrates that LDA-Reg is especially beneficial for applications where training data is insufficient. In this case, external knowledge can be used via LDA-Reg to complement limited training data and alleviate overfitting.

7.5.2 Comparison with Word Embedding Methods

In this section, we compare LDA-Reg with state-of-the-art word embedding methods on the MIMIC-III dataset and the Sentence Polarity dataset for both one layer and two layer models. Since both LDA-Reg and word embedding methods are able to make use of external knowledge, we do not need to investigate the effectiveness of incorporating external knowledge by varying the amount of training data. Consequently, we use all the available training data and report the results.

Figures 5, 6 and 7 show the AUC for different methods on all the datasets for both one layer and two layer models. For SKG-Topic-Add and SKG-Topic-Concat, these two word embedding methods incorporate information of the topic model, i.e., element addition and concatenation of the topic-based word embedding and skip-gram, so that both the semantic information of local windows and global statistical information are captured.

From the figures, we observe that SKG-Topic-Add and SKG-Topic-Concat are not dominating the other word embeddings methods. The reason is that the topic-based word embeddings are only trained using unique words, which leads to much less training data than the original word embedding methods that consider all words. LDA-Reg achieves better results than all the baseline word embedding methods. This is attributed to the fact that LDA-Reg is able to incorporate external knowledge into hidden neurons. Consequently, the knowledge is made an integral part of the training process and the incorporation of the external knowledge can be adaptive to the prediction task. In comparison, for word embedding methods, the learned word vectors are fed as the input to the NN model. Thus, it is not able to incorporate knowledge adaptively to the training task. For SKG-Topic-Add and SKG-Topic-Concat, although these two word embedding methods contain topic information, such information is embedded in the embedding vectors and is fed as the input to the NN model like other word embedding methods, where the use of topic information is not as deep into neurons and adaptive as LDA-Reg and hence leads to worse results than LDA-Reg.

Furthermore, we note that the improvement of LDA-Reg over baseline word embedding methods is more obvious in the MIMIC-III dataset. This is because MIMIC-III is a much more difficult task, i.e., the MIMIC-III dataset's input features are more complex, including features from different sources, e.g., diagnoses, lab tests and medications, etc., leading to the complicated relationship among the input features. LDA-Reg takes advantage of word mixtures Φ learned from the external corpus to organize different input features into semantic groups and help hidden neurons attend to different subsets of input features by imposing customized adaptive regularization, which is beneficial for dealing with the complex multi-source input features.

TABLE 1
Top Five Topics of Knowledge-Related Neurons on MIMIC-III Dataset

Neuron ID: 3	Neuron ID: 17	Overall Salient Topics
Renal Failure	Vitamin Deficiency	Renal Failure
Cell Related LabTest	Cholecalciferol Deficiency	Cell Related LabTest
Coronary Artery Disease	Hematology LabTest	Abnormal Weight Gain
Alkaline Phosphatase	Sepsis	Alkaline Phosphatase
Abnormal Weight Gain	Alkaline Phosphatase	Prinzmetal angina

TABLE 2
Top Five Topics of Knowledge-Related Neurons on Sentence Polarity Dataset using LMRD

Neuron ID: 36	Neuron ID: 42	Overall Salient Topics
Urban	Crime/Suspense	Crime/Suspense
Music	Urban	Urban
Crime/Suspense	Male	Music
Female	Sports	Male
Male	Music	Female

In this dataset, LSTM performs better than MLP. The reason is that the Sentence Polarity dataset aims to predict the sentiment label of a sentence according to the word sequence of the sentence and the sequence length of the Sentence Polarity dataset is 25, much longer than that of the MIMIC-III dataset. LSTM, which is designed for dealing with the sequential input data and taking advantage of the historical information of long sequences, is advantageous in this dataset.

7.6 Interpretation

One advantage of LDA-Reg is its ability to explain knowledge-related neurons in the neural network models. In this section, we show both global model interpretability by inspecting topic mixture parameters and local model interpretability for individual samples. Specifically, for global model interpretability, we show the topics of representative knowledge-related neurons. In terms of local interpretability, we show how LDA-Reg identifies the customized significant topics for individual patients in the disease prediction task. For these experiments, we use the one layer MLP model. Also, we use LMRD as the external corpus for the Sentence Polarity dataset. However, the interpretation method introduced here can also be applied on different NN models as well as other corpora.

7.6.1 Global Model Interpretation

Topics of Knowledge-related Neurons. Through the topic mixture parameters, we are able to obtain the salient topics of a knowledge-related neuron. In tables 1 and 2, we rank the topics of the knowledge-related neuron according to topic mixture parameters and show the results. The first two columns of these two tables show the top five topics for two representative knowledge-related neurons. From table 1, we can find that for the MIMIC-III dataset, the top topics of a knowledge-related neuron are typically related or are comorbidities. For example, the neuron of coordinate ID 3 is about renal failure and its comorbidity, coronary artery disease, which is a common type of heart disease. For the neuron of coordinate ID 17, cholecalciferol deficiency

TABLE 3
Representative Topics for Disease Prediction on MIMIC-III Dataset

Patient 1	**Congestive Heart Failure (ICD9-428.0)** **Venous Catheterization(ICD9-3893)**
Neurons ID	**Representative Topics**
42	Myocardial Infarction Medication
93	Coronary Artery Disease
96	Renal Failure
Patient 2	**Hypertensive Chronic Kidney Disease(ICD9-403.91)** **Hemodialysis(ICD9-3995)**
Neurons ID	**Representative Topics**
64	Renal Disease & Medication
93	Coronary Artery Disease
127	Renal Disease LabTest

is a subtype of vitamin deficiency. Table 2 shows the top topics for the Sentence Polarity dataset. The neuron of coordinate ID 36 is related to more relaxing topics such as urban and music, while the neuron of coordinate ID 42 is more tensional because the crime/suspense topic ranks the highest. By inspecting the salient topics of each knowledge-related neuron, we are able to know what a knowledge-related neuron ”means”. This is an innovative approach to explaining what the hidden neuron tries to capture in the NN models.

Apart from the topics of representative knowledge-related neurons, we are also interested in the topics that are salient over all knowledge-related neurons. We calculate the sum of topic mixture parameters over all knowledge-related neurons and then denote the topics with the highest sums as ”Overall Salient Topics” shown in the third column of these two tables. These topics can be understood as significant topics for the prediction task.

7.6.2 Local Model Interpretation

Risk Factors for Disease Prediction. Given supervised tasks, LDA-Reg is able to identify customized significant topics for each sample. In this section, we take the disease prediction task on the MIMIC-III dataset, which requires healthcare domain knowledge, as an example. We show interpretable risk factors found by the LDA-Reg for specific patients and verify them by doctors from the hospital.

The interpretation process takes two steps. **Firstly** we identify representative neurons for each patient by taking advantage of the gradient-based method [29]: after the model converges, for each patient, we obtain the significance of each knowledge-related neuron by calculating its gradient with respect to the loss function. The knowledge-related neurons which have high significance values are regarded as representative neurons for this patient. **Secondly**, for each representative neuron, we obtain the most representative topic according to the topic mixture parameters as introduced in Section 7.6.1.

We take two patients as an example. Table 3 shows the interpretation results. In Table 3, **Patient 1** is diagnosed with two diseases, namely congestive heart failure and venous catheterization. The top three representative neurons are neurons of coordinate IDs 42, 93 and 96. The most representative topics for these three neurons are myocardial infarction medication, coronary artery disease and renal failure respectively. With the help of doctors from NUHS, we verified that the found topics are closely related to the diseases patient 1 is diagnosed with. Specifically, with regard to congestive heart failure, coronary artery disease is a related heart disease and renal failure is a typical comorbidity. **Patient 2** is a patient with kidney disease and is getting hemodialysis. The representative topics of the top three representative neurons are all related to kidney disease. Specifically, the topics of neurons of coordinate IDs 64 and 127 are renal disease, medication and lab test which are closely related to kidney disease while the topic of neuron of coordinate ID 93, coronary artery disease, is a comorbidity of kidney disease. It should be noted that this interpretation process is general to different kinds of NN models with different numbers of hidden layers. As long as the representative knowledge-related hidden neurons are identified, LDA-Reg is able to interpret these neurons with representative topics.

Clustering of Knowledge-related Neurons. In this section, we evaluate the interpretability of LDA-Reg in a quantitative way using clustering. Specifically, we show the interpretability of the learned topic mixture parameters of each hidden neuron by exploring the association between neurons and labels. The interpretation process takes three steps. **Firstly**, for each disease a patient is diagnosed with, we identify the most representative neuron using the gradient-based method and assign this disease to this neuron as the label. We go through all the patients so that neurons are assigned with labels. Note that one neuron may be assigned with multiple labels. **Secondly**, we take advantage of the spectral clustering method [8] to cluster neurons using the topic mixture parameter vectors. Cosine similarity is utilized as the distance metric for the clustering. **Lastly**, after the first two steps, each neuron is associated with labels and assigned to a cluster. To evaluate the performance of clustering, we employ the class entropy as the evaluation metric [1]. Class entropy measures the homogeneity of labels and lower values of class entropy indicate that neurons of the same label are assigned to fewer clusters, which means better performance. We experiment with # clusters equaling 2, 5, 10 and 20 and # topics equaling 20, 50, 100, 200 and 500. We then average the class entropy over different # clusters for each # topics. The averaged results and standard deviation are reported in Figure 8. It can be observed that # topics equaling 20 obtains the highest class entropy, which is due to the fact that such small number of topics does not capture the semantic information well enough. When # topics is larger than 50, the class entropy is slightly increasing, which indicates # topics does not affect the interpretability of topic mixture parameter vectors significantly when # topics is relatively large.

7.7 Effectiveness of Hyperparameters

As shown in Equation 10, both contribution strength λ and Dirichlet hyperparameter $\overrightarrow{\alpha}$ affect the learning of topic mixture parameters. In this section, we are therefore interested in investigating the effects of λ and $\overrightarrow{\alpha}$. For all the experiments, we use one hidden layer models. Also, we use LMRD as the external corpus for the Sentence Polarity dataset.

[8] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html

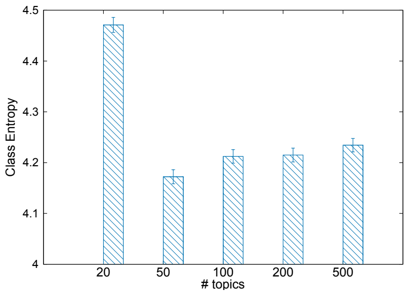
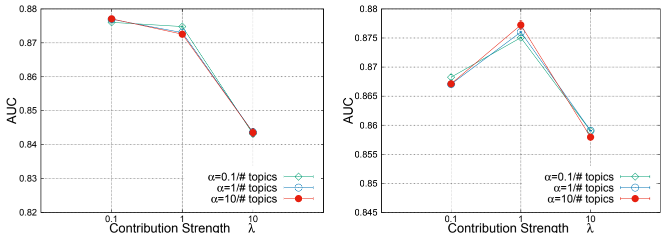


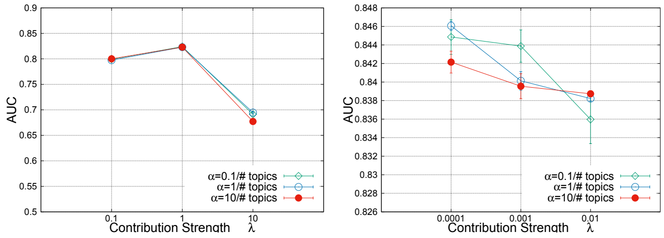
Fig. 8. Performance on clustering of knowledge-related neurons.

In our experiment, we set every dimension of $\overrightarrow{\alpha}$ the same value and use α to denote the value of each dimension. This value is set to be inversely proportional to # topics, which is 50 for MIMIC-III and 200 for Sentence Polarity using LMRD corpus. Figures 9 and 10 show the AUC for different combinations of λ and α for both datasets. The result shows that, for both datasets, $\overrightarrow{\alpha}$ does not have significant impact on the AUC. This suggests that hyperparameter $\overrightarrow{\alpha}$ does not need to dominate in Equation 10. In terms of contribution strength λ, we observe that λ equaling 10 achieves the worst results. This is due to the fact that large λ incurs strong regularization, which is harmful to the model.



(a) MLP (b) LSTM

Fig. 9. AUC for different α and λ values for MIMIC-III dataset.



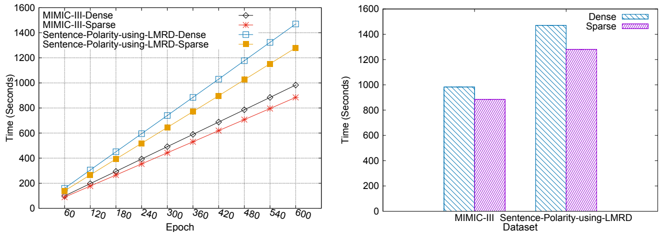
(a) MLP (b) LSTM

Fig. 10. AUC for different α and λ values for Sentence Polarity dataset using LMRD corpus.

7.8 Effectiveness of Sparse Update

In this section, we evaluate the effectiveness of our proposed sparse update method. Figure 11(a) shows the training elapsed time with respect to epochs for dense and sparse update methods with one layer MLP model in two datasets. Here, dense update refers to the update method that does not employ the sparse update method. For both datasets, we can see that the training elapsed time for the sparse update method is less than that of the dense update method. Also, the training elapsed time for the sparse update method grows linearly in time as the number of epochs increases, which proves the effectiveness of sparse update. Figure 11(b) shows the total time of sparse and dense update method for the two datasets. In both datasets, the total time of the sparse update method is less than that of the dense update method, which is consistent with the observation in Figure 11(a) and further confirms the effectiveness of our proposed sparse update method.

Table 4 shows the memory consumption for dense and sparse update in two datasets. From the table, we observe that the input data of dense update (one-hot representation) is extremely sparse, i.e., only 2.28% (1533741/67327374) and 0.28% (106613/37837044) cells have data for MIMIC-III and Sentence Polarity datasets respectively. For the sparse update, after transforming the input data to the sparse format, the memory consumption for the two datasets is only 3.43% (17.61/513.67) and 0.43% (1.25/288.67) that of the dense update, which reduces the space complexity by a wide margin. In addition, since the calculation of sparse update is exactly the same as dense update, there is no drop in the model performance for the sparse update.



(a) (b)

Fig. 11. Comparison between sparse update and dense update.

TABLE 4
Comparison of Memory Consumption for Input Data

MIMIC-III		
Method	**# cells**	**Memory Consumption (MB)**
Dense Update	67327374	513.67
Sparse Update	1533741	17.61
Sentence-Polarity-using-LMRD		
Method	**# cells**	**Memory Consumption (MB)**
Dense Update	37837044	288.67
Sparse Update	106613	1.25

7.9 Effectiveness of Lazy Update

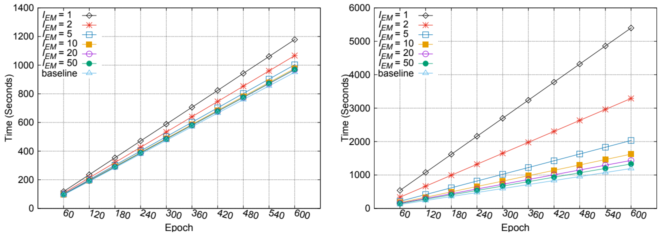
For the lazy update, the update interval I_{EM} as well as first epochs hyperparameter E are significant hyper-parameters. In this section, we investigate the effects of these hyperparameters in terms of computational time with one layer MLP model in two datasets.

7.9.1 *Performance of Update Interval* I_{EM}

Figure 12 shows the training time with respect to the number of epochs for the baseline (L2-Reg) and the lazy update algorithm with different I_{EM} values. It can be observed that the time of algorithms with different I_{EM} values linearly grow when the number of epochs increases, which confirms the effectiveness of the lazy update algorithm. Among all six I_{EM} values, the algorithm without the lazy update (I_{EM} = 1) takes the longest time and the algorithm with I_{EM} = 50 takes the shortest. This is due to the fact that the algorithm with a larger I_{EM} updates responsibilities and topic mixture parameters less frequently. Specifically, the computational time of algorithm with I_{EM}=50 is nearly the same as that of the baseline method, without a drop in AUC. This again confirms the effectiveness of the proposed lazy update algorithm.

7.9.2 Performance of First Epochs Hyperparameter E

Figure 13 shows the training time with respect to epochs for the baseline (L2-Reg) and the lazy update algorithm with different E values. Since the first epochs hyperparameter E affects the early stage of training, we only show the first 60 epochs' results. From the figures, we observe that at epochs 5, 10, 20, algorithms with different E values diverge. This is because the lazy update algorithm spends more time computing responsibilities and topic mixture parameters each epoch before E epochs. The effect of different E values of the Sentence Polarity dataset is more obvious than that of the MIMIC-III dataset. This is because the number of topics for the Sentence Polarity dataset is 200, much larger than that of MIMIC-III dataset, which is 50. The larger number of topics causes a longer training time each epoch when the lazy update is not employed, which leads to the larger training time difference between the first E epochs and the remaining epochs. After 60 epochs, the algorithm takes the most time when E=50 and takes the least when E=1. This is because the algorithm with larger E takes more time in the first E epochs when the lazy update is not employed. When E is set to 1, the training time of LDA-Reg is nearly the same as the baseline, without an AUC drop.

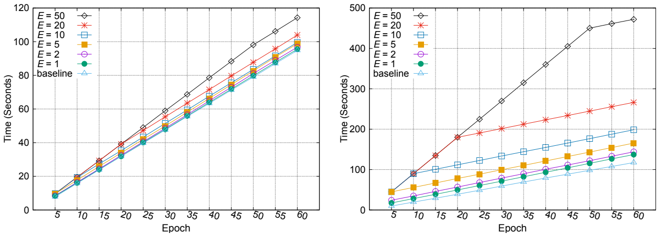


(a) MIMIC-III (b) Sentence-Polarity-using-LMRD

Fig. 12. Time for different update interval values.

8 Conclusions

In this paper, we propose a novel knowledge driven regularization framework, LDA-Reg, to incorporate external knowledge from unstructured corpora into the NN model. Efficient update method EM-SGD that incorporates EM and SGD is designed to update topic mixture parameters and model parameters. Lazy update and sparse update algorithms are also devised for the high-dimensional inputs and sparse inputs respectively. Experiments show that LDA-Reg obtains better performance than the regularization method that does not incorporate external knowledge. Also, our LDA-Reg yields better performance than existing state-of-the-art word embedding methods while providing meaningful interpretation for the hidden neurons of NN models.



(a) MIMIC-III (b) Sentence-Polarity-using-LMRD

Fig. 13. Time for different E values.

In this work, the external knowledge is integrated into the first layer. Moving forward, we plan to extract external knowledge with the hierarchical LDA and then integrate the knowledge into hidden layers. Another extension of the work would be constructing a knowledge graph using the external corpora for integrating into the NN model.

Acknowledgments

This research is supported by the National Science and Technology Major Project (No. 2018ZX10201002) and the National Research Foundation Singapore under its AI Singapore Programme (Award Number: AISG-GC-2019-002).

References

[1] J. Bakus, M. Hussin, and M. Kamel. A som-based document clustering using phrases. In *Proceedings of the International Conference on Neural Information Processing, 2002.*, volume 5, pages 2212–2216, 2002.

[2] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[3] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *Journal of machine Learning research*, 3:1137–1155, 2003.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[5] Z. Chen, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh. Discovering coherent topics using general knowledge. In *Proceedings of the International Conference on Information and Knowledge Management*, pages 209–218, 2013.

[6] M. Cogswell, F. Ahmed, R. Girshick, L. Zitnick, and D. Batra. Reducing overfitting in deep networks by decorrelating representations. *arXiv preprint arXiv:1511.06068*, 2015.

[7] R. Das, M. Zaheer, and C. Dyer. Gaussian lda for topic models with word embeddings. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 795–804, 2015.

[8] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391, 1990.

[9] M. Gaur, U. Kursuncu, and R. Wickramarachchi. Shades of knowledge-infused learning for enhancing deep learning. *IEEE Internet Computing*, 23(6):54–63, 2019.

[10] S. Gerard and J. M. Michael. Introduction to modern information retrieval, 1983.

[11] J. Han, J. Pei, and M. Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[12] G. Heinrich. Parameter estimation for text analysis,"univ. leipzig, leipzig. Technical report, Germany, Tech. Rep., http://faculty. cs. byu. edu/˜ ringger/CS601R/papers/Heinrich-GibbsLDA. pdf, 2008.

[13] T. Hofmann. Probabilistic latent semantic indexing. In *ACM SIGIR Forum*, pages 211–218, 2017.

[14] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 2016.

[15] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[16] T. Kneib, S. Konrath, and L. Fahrmeir. High dimensional structured additive regression models: Bayesian regularization, smoothing and predictive performance. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 60(1):51–70, 2011.

[17] A. Krogh and J. A. Hertz. A simple weight decay can improve generalization. In *Advances in Neural Information Processing Systems*, 1991.

[18] K. Lund and C. Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior research methods, instruments, and computers*, 28(2):203–208, 1996.

[19] Z. Luo, S. Cai, G. Chen, J. Gao, W.-C. Lee, K. Y. Ngiam, and M. Zhang. Improving data analytics with fast and adaptive regularization. *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[20] Z. Luo, S. Cai, C. Cui, B. C. Ooi, and Y. Yang. Adaptive knowledge driven regularization for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[21] Z. Luo, S. Cai, J. Gao, M. Zhang, K. Y. Ngiam, G. Chen, and W.-C. Lee. Adaptive lightweight regularization tool for complex analytics. In *International Conference on Data Engineering*, pages 485–496, 2018.

[22] Z. Luo, S. H. Yeung, M. Zhang, K. Zheng, G. Chen, F. Fan, Q. Lin, K. Y. Ngiam, and B. C. Ooi. Mlcask: Efficient management of component evolution in collaborative data analytics pipelines. In *International Conference on Data Engineering*, 2021.

[23] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 142–150, 2011.

[24] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[25] B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2005.

[26] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1532–1543, 2014.

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[28] T. Scheepers, E. Kanoulas, and E. Gavves. Improving word embedding compositionality using lexicographic definitions. In *International Conference on World Wide Web*, pages 1083–1093, 2018.

[29] R. R. Selvaraju, P. Chattopadhyay, M. Elhoseiny, T. Sharma, D. Batra, D. Parikh, and S. Lee. Choose your neuron: Incorporating domain knowledge through neuron-importance. In *Proceedings of the European conference on computer vision*, pages 526–541, 2018.

[30] A. Srivastava and C. Sutton. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*, 2017.

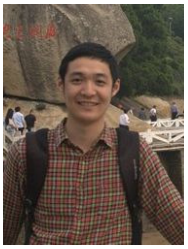
[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[32] H. Xu, M. Dong, D. Zhu, A. Kotov, A. I. Carcone, and S. Naar-King. Text classification with topic-based word embedding and convolutional neural networks. In *Proceedings of the International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 88–97, 2016.

[33] M. Zhang, C. R. King, M. Avidan, and Y. Chen. Hierarchical attention propagation for healthcare representation learning. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pages 249–256, 2020.

[34] P. Zhang, S. Wang, D. Li, X. Li, and Z. Xu. Combine topic modeling with semantic embedding: Embedding enhanced topic model. *IEEE Transactions on Knowledge and Data Engineering*, 2019.

Kai Yang received the BEng degree from School of Computer Science and Technology from Nanjing University of Posts and Telecommunications, China in 2014. He is currently a PhD student of School of Electronic Engineering and Computer Science at the Peking University. His research interests include the area of deep learning, healthcare analytics and knowledge graph.



Zhaojing Luo received the BEng degree from School of Computer Science and Technology from Huazhong University of Science and Technology, China in 2014. He is currently a PhD student of School of Computing at the National University of Singapore. His research interests include the areas of deep learning, machine learning and healthcare analytics.

Jinyang Gao received the BSc degree from School of Electronics Engineering and Computer Science from Peking University, China, and the PhD degree in computer science from the National University of Singapore, in 2012 and 2016, respectively. He is currently a research fellow of School of Computing at the National University of Singapore. His research interests include the areas of database, deep learning, machine learning.

Junfeng Zhao is an associate professor in the Software Institute, School of Electronics Engineering and Computer Science, Peking University. Her research interests include Big Data Analysis, Software Engineering and Knowledge Engineering, Software Reuse and Component Technology. Dr. Zhao has published more than 40 research papers, and most of them are published in High rank conferences, such as AAAI, ICSR, and ICDM. She has presided 8 national technical research projects including NSFC, 863 projects, etc. She also took in charge of 5 Provincial and ministerial level projects. She was awarded Second prize of national science and technology progress award (Rank Fifth) in 2006 and Second prize of Beijing science and technology progress award (Rank Fifth). She is the Secretary General of “Big Data Techniques Standardization Group” of China national information Technology Standardization. She is a member of the IEEE.

Beng Chin Ooi is currently a distinguished professor of Computer Science at the National University of Singapore. His research interests include database system architectures, performance issues, indexing techniques and query processing, in the context of multimedia, spatio-temporal, distributed, parallel, blockchain, and in-memory systems. He served as the editor-in-chief of the IEEE Transactions on Knowledge and Data Engineering (2009-2012), a trustee board member and the president of the VLDB Endowment (2014-2017). He is a fellow of the IEEE, ACM and Singapore National Academy of Science. He is serving as the editor-in-chief of ACM Transactions on Data Science.

Bing Xie received the Ph.D degree in Computer Science from National University of Defense Technology in 1998. He is a professor and doctoral supervisor at Peking University since 2007. His research interests include software engineering,formal methods, knowledge engineering, etc. He is a member of the IEEE.