RESEARCH ARTICLE

Biomedical literature classification with a CNNs-based hybrid learning network

Yan Yan[1]*, Xu-Cheng Yin[2]*, Chun Yang[2], Sujian Li[3], Bo-Wen Zhang[2]

1 Department of Computer Science and Technology, School of Mechanical Electronic and Information Engineering, China University of Mining and Technology, Beijing, 100083, China, **2** Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China, **3** Key Laboratory of Computational Linguistics, Peking University, Ministry of Education, Beijing 100871, China

* yanyanustb@126.com (YY); xuchengyin@ustb.edu.cn (XCY)

Abstract

Deep learning techniques, e.g., Convolutional Neural Networks (CNNs), have been explosively applied to the research in the fields of information retrieval and natural language processing. However, few research efforts have addressed semantic indexing with deep learning. The use of semantic indexing in the biomedical literature has been limited for several reasons. For instance, MEDLINE citations contain a large number of semantic labels from automatically annotated MeSH terms, and for a great deal of the literature, only the information of the title and the abstract is readily available. In this paper, we propose a Boltzmann Convolutional neural network framework (B-CNN) for biomedicine semantic indexing. In our hybrid learning framework, the CNN can adaptively deal with features of documents that have sequence relationships, and can capture context information accordingly; the Deep Boltzmann Machine (DBM) merges global (the entity in each document) and local information through its training with undirected connections. Additionally, we have designed a hierarchical coarse to fine style indexing structure for learning and classifying documents, and a novel feature extension approach with word sequence embedding and Wikipedia categorization. Comparative experiments were conducted for semantic indexing of biomedical abstract documents; these experiments verified the encouraged performance of our B-CNN model.

OPEN ACCESS

Citation: Yan Y, Yin X-C, Yang C, Li S, Zhang B-W (2018) Biomedical literature classification with a CNNs-based hybrid learning network. PLoS ONE 13(7): e0197933. https://doi.org/10.1371/journal.pone.0197933

Editor: Constantino Carlos Reyes-Aldasoro, City University London, UNITED KINGDOM

Received: January 13, 2017

Accepted: May 10, 2018

Published: July 26, 2018

Data Availability Statement: All relevant data are from third-party sources and are available at the following URLs: http://muchmore.dfki.de/resources1.htm; http://biotext.berkeley.edu/data.html; http://www.bioasq.org/participate/challenges; http://qwone.com/~jason/20Newsgroups/.

Funding: This work was partly supported by the National Natural Science Foundation of China (61175020) by Xu-Cheng Yin. There was no additional external funding received for this study.

Competing interests: The authors have declared that no competing interests exist.

1 Introduction

With the rapid development of biomedicine, there has been an increase in the amount of biomedical literature. Semantic indexing is vital for biomedicine document classification and retrieval. Semantic indexing for biomedical documents is multi-label classification problem to assign the articles with controlled vocabulary thesaurus, MeSH (Medical Subject Heading) terms. Table 1 shows an example of a biomedical article with manually annotated MeSH terms. The annotation work would consume immense manpower and financial resources. The support for retrieving newly-published articles would also be delayed. The main purpose of the task is to automatically classify new PubMed documents and approximate the results with the

Table 1. An example PubMed article with manually annotated MeSH terms.

Journal: Photochemistry and photobiology
Year: 1983
Title: Kinetics of bacterial bioluminescence and the fluorescent transient.
Abstract: The addition of FMNH(2), to Vibrio harveyi luciferase at $2^{\circ}C$ in the presence of tetradecanal results in the formation of a highly fluorescent transient species with aspectral distribution indistinguishable from that of the bioluminescence. The bioluminescence reachesmaximum intensity in 1.5 s and decays in a complex manner with exponential componentsof 10(-1) s(-1), 7 x 10(-3)S(-1). and 7 x10(4)s(-1). The fluorescent transient rises exponentially at7 x 10(-2)s(-1) and decays at 3 x 10 (4)s(-1). The slowest bioluminescence component. comprising the bulk of the bioluminescence. decays at twice the rate of the fluorescent transient under all variations of reaction conditions: concentration of reactants. temperature 2–$20^{\circ}C$. and aldehyde chain length - decana1, dodecanal and tetradecanal. The activation energy for both the slowest bioluminescence decay and the transient fluorescence decay is 80 kJ-mol(-1). An energy transfer scheme is proposed to explain the results where two distinct chemically energized species utilize the fluorescent transient as emitter for the slower bioluminescences, and for the faster process a fluorophore present in the protein preparation. Kinetic observations suggest that typical preparations of V. harveyi luciferase comprise 15% active protein.
MeSH terms: “Flavin Mononucleotide” “Fluorescence” “Kinetics” “Luciferases Luminescence” “Time Factors” “Vibrio”

https://doi.org/10.1371/journal.pone.0197933.t001

manually annotated classes from MeSH headings provided by Pubmed curators. As illustrated, the original text of article title as well as abstract are given. Over 20 thousand descriptors in MeSH headings are provided as candidates of which one or some are associated with the article. There are two major research focuses with respect to semantic indexing [1–3]. The first is based on shallow learning approaches, which usually contain a process of matching correlated documents by comparing keywords. This is the most primitive and direct method. The other is based on deep neural networks (DNNs) [4], which have emerged as a powerful machine learning technology. DNNs have been widely applied and proved to be effective in image classification [5–7], speech recognition [8–10], and natural language processing (NLP) tasks [11–13]. Through several comparisons, they are proved to have significant advantages compared to state-of-the-art shallow learning methods for semantic indexing [14–16].

In 2011, researchers from Microsoft Research and Google used DNN technology for speech recognition to decrease the error rate of 20% -30% [14]. In 2012, DNN technology was utilized on ImageNet task [16], which reduce the error rate from 26% to 15%.

In the field of biomedicine, MeSH indexing, the mostly applied mainstream method, mines the context of indexed PubMed and MEDLINE citations [17–21] based on the bag-of-words model (BOW). Nowadays, several statistical models and language models in information retrieval and text mining are introduced to combine with MeSH mapping [22–25]. However, this type of feature representation only contains information related to word frequency. In many common area cases like sentiment classification, deep learning has been effectively used to represent more semantic information other than frequency. However, in biomedicine, conventional deep learning methods might face some more challenges, such as terminologies, synonyms based on professional knowledge [26, 27]. The MeSH term “Luciferases Luminescence” in the example within Table 1 cannot be easily mapped without professional knowledge. Indeed, indexing is hindered by a large quantity of professional terms, a lack of information (generally only titles and abstracts, which may cause data sparsity for learning), and a high correlation between different labels. Hence, few research efforts have addressed for indexing with deep learning.

Convolutional neural networks (CNNs) [28, 29] provide a flexible framework that can be used to reduce variation and exploit spatial correlations using weight sharing and local connectivity. However, the sizes of the sliding windows of the convolutional kernels can be difficult to determine. Large windows can hinder training because they necessitate an enormous number of parameters [30]. Conversely, small window sizes may lead to the loss of some critical information, which then needs to be estimated. The deep Boltzmann machine (DBM) [31] represents a good feature extraction method in deep learning, as it can be used to effectively combine global and local information. In this paper, taking biomedical abstract semantic indexing as a case study, we propose a hierarchical CNNs-based (coarse to fine) indexing framework and present a suitable loss function for handling specific domain terms and the correlation of labels for hierarchical indexing of biomedical documents. As indexing must be accomplished using only the title and abstract information of documents, we used word sequence embedding together with Wikipedia categories and entity classes to enrich document representation. Our empirical results verify that this improved representation is denser than that obtained via the BOW model.

To summarize, this paper represents three major contributions to the semantic indexing literature. First, to the best of our knowledge, we present the first case study of biomedical document semantic indexing with CNNs, including comparisons with several state-of-the-art methods. Second, we offer a framework for combining CNNs with DBM to learn document representation, where CNNs extract local information regarding the context and the DBM is used to merge the global features. We use a suitable loss function for the training of this framework, where multi-label classification is performed in a coarse-to-fine learning style. Third, we propose a new way to enrich document representation using sequence information.

This paper is organized as follows. Related work is described in Section 2. In Section 3, we describe the proposed method for semantic indexing with CNNs. In Section 4, we describe semantic feature extension for a document. We demonstrate comparative experiments in Section 5 and present final remarks in Section 6.

2 Related work

2.1 Semantic indexing for documents

The existing semantic indexing approaches can be divided into two main categories, according to the semantic representation methods. One group of methods tend to represent semantics with shallow artificial features, like the BOW and term frequency-inverse document frequency (TF-IDF) models, which only contain information about word frequency. However in most cases, the corresponding lexical matching would cause missing results because of different expressions of significant concepts, for example aliases, synonyms or abbreviations. Several learning methods have been proposed with attempt based on the shallow semantic features, such as latent semantic indexing (LSI), latent Dirichlet allocation (LDA) and probabilistic latent semantic indexing (pLSI) [32–34]. LSI aims at decomposing the feature matrices with SVD (Singular Value Decomposition), to select a subset of the original features to represent the semantics. The process is similar to dimension reduction with Principle Component Analysis. Through the removal process, which is implemented by assigning weights to 0, the unimportant latent semantic features are filtered out. A vital weakness of this method is the inability of processing the problem of polysemy. pLSI constructs a semantic representation model in which each document uses massive parameters. Consequently, when processing large number of documents, the overfitting problem are unavoided. In order to solve the above limitation, LDA was proposed to eliminate the dependency between the number of parameters in the model and the number of documents. However, most of these above models are linear models with unsupervised dimension reduction [35], and are based on the unordered hypothesis (exchangeability). That is, the sequence of the words in the document is always ignored; thus, the semantic representation is inappropriate for some sequences [3]. Recently, some researchers have attempted to apply supervised learning techniques to optimize the semantic representation of documents. Typically, supervised LDA [36] adds an extra response variable to LDA by generalizing linear models with respect to the EM (Expectation Maximization) algorithm associated with each document. However, the query and the document are regarded as independent and processed separately in the learning process. Supervised semantic indexing [37] focuses on the correlations between words in queries and candidate documents, and learning to rank algorithms are then utilized to select the best combination of features from a large feature set generated from all word pairs.

In the last several years, deep learning has arisen the focus of all researchers in the research field of artificial intelligence, with a variety applications on image classification [16], object recognition [38, 39], speech recognition [14], and NLP tasks [15, 40]. Like other applications, there are some investigations about trying deep learning techniques for semantic indexing. Salakhutdinov and Hinton [1] proposed a novel representation method for extending semantic indexing. An deep auto-encoder model was proposed in that work with the combination of a higher layer encoded with binary codes and a lower layer generated based on word frequency vectors. The constrained Poisson model was also introduced to deal with documents with various lengths. Another work was proposed by Mirowski [2] to improve the deep auto-encoder model by introducing a dynamic variable. This variable is generated from a gradient-based MAP (Maximum a Posteriori) inference and can help to compute the encoder and the decoder. With an additional use of the document label, the classifiers can be successfully trained. Socher [41] proposed a semantic indexing method with the recursive neural network, which captures the semantics of sentences via a tree structure for labeled structure prediction. This model has been found to be efficient for constructing sentence representations when mapping to images. Wu [3] designed a deep structure with restricted Boltzmann machines (RBMs) to compute semantic representation of documents. This model was improved in [42] which develops the feature space can make labels less interdependent and easier to model and predict at inference time. CNN was firstly utilized in semantic indexing in [43] with dynamic max pooling and hidden bottleneck layers. There are also several studies working for semantic indexing for images, videos, etc [44–46]. Through experiments in the above papers, deep semantic embedding models with nonlinear features can grasp the semantics with more accurate and robust representations. These models also use discriminative fine-tuning, and improve the ranking of documents with more appropriate scoring functions. Indeed, deep models are associated with an increase in indexing performances.

2.2 Biomedicine document classification

Several studies have been investigated natural language processing techniques in the health domain [47–49]. MTI (Medical Text Indexer) [20] is a document indexing system that provides recommendations based on the Medical Subject Headings (MeSH) and MEDLINE databases. MTI has two main components: MetaMap Indexing and PubMed Related Citations (PRC). The MetaMap is a module that maps biomedicine documents to concepts in the UMLS Metathesaurus. The PRC algorithm is a modified kNN algorithm that relies on document similarity to assign MeSH headings. The results produced by the two paths are given weights via post-processing (such as clustering and merging), and these can be used to retrieve a ranked list of MeSH terms.

The MeSHUP system [21] explores the combination of different machine learning approaches for classification of full class-sets. This system is highly scalable and capable of improving biomedical IR from a ranked output of MeSH terms.

2.3 CNNs for document classification

Lai [30] proposed the recurrent convolutional neural networks (RCNN) model for document classification, with the goal of overcoming the Recurrent Neural Network bias whereby later words are more dominant than earlier words. The RCNN effectively utilizes the advantages of the recurrent structure, which captures the contextual information, and learns the feature representation of documents using CNN. Shen [50] proposed a new latent semantic representation for web searches based on CNN. The word n-gram (word sequence) information is set as the local feature and is calculated via convolution and max pooling to learn the high-level semantic representation. Johnson [51] explored the application of CNN for converting images (2D structure) to text (1D structure). In documents, the word order is entered as a 1D structure, and each word representation uses a vector based on the BOW. Experiments indicated that this model achieved good performance with respect to sentiment classification and topic classification. Santos [26] proposed a deep convolutional neural networks model using character-level, word-level, and sentence-level features for sentiment analysis of short texts. These features are not based on handcrafted inputs but on word-level embedding. Additionally, these features, which are operated by a convolutional layer, are able to capture the global feature vectors of sentences and extract the relevant features from any part of a word.

3 Model

Considering the numerous classes of documents and the imbalance in the distributed samples, we developed a framework in which hierarchical CNNs were combined with DBMs. This framework can perform multi-class and multi-label semantic indexing with correlated labels for biomedicine documents (we call this B-CNN). The architecture of our proposed framework is shown in Fig 1.

The model consists of three parts, i.e., feature representation (the input of the framework), the B-CNN model, and multi-label hierarchical classification (the output of the framework). In this section, we mainly describe the B-CNN model and the components of hierarchical indexing. B-CNN combines the advantages of both CNNs and DBMs. CNNs are more suitable for capturing context information from features with sequence relationships. DBMs are advantageous in that they can merge global and local information through undirected connections, thus effectively removing noise from different representations. The hierarchical B-CNN based indexing is much better than flat classification for processing a large number of classes. Moreover, the coarse clustering step is an effective way to remove noise from unbalanced distributed samples. In addition, we designed suitable loss functions for the learning process of this framework. The details of these two parts are described as follows.

3.1 Convolutional neural networks

Since the error gradients and the BP (Back Propagation) algorithm into the CNNs to simlify the training process [28, 29], CNNs become more widely-used for feature representations. A simple CNN structure acting as sentence model is made up with one-dimensional convolutional layer followed by a k-max pooling layer (see the CNN model in a single Fig 1). Different from conventional neural networks, each neuron in convolutional layer is only connected with a local subset of the neurons in the lower layer. In this way, the neuron merely aims at extracting the local information of the corresponding lower layer. Each neuron represents local features from the connecting receptive field (which may or may not overlap), depending on the activation function within the neuron. As a result, convolution neurons play the role of feature detectors, and the weights of the neural connections decide the characteristics of the reaction (i.e., the degree of the reaction). Through feature detectors, several feature maps are generated

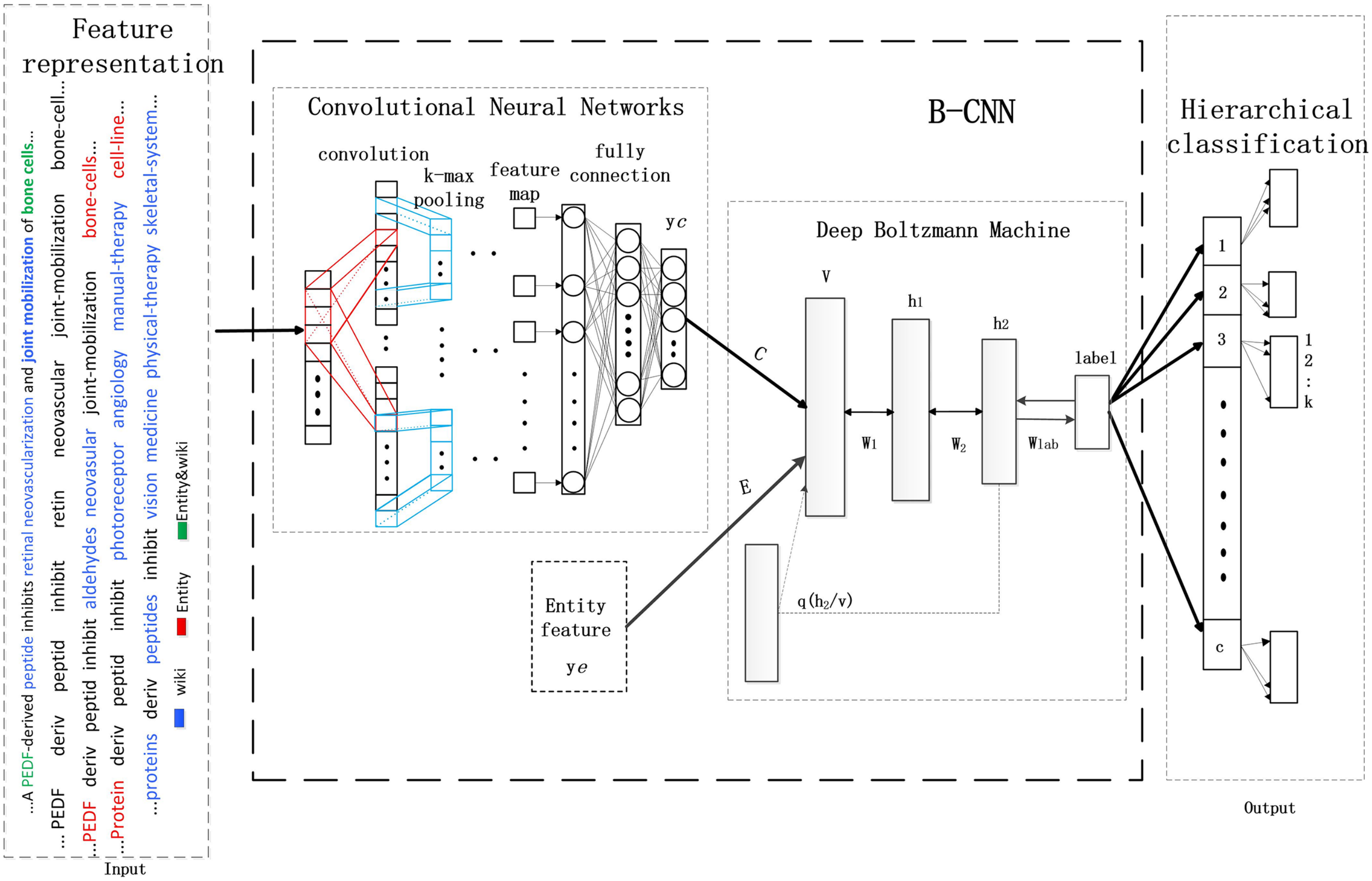
Feature representation
...A PEDF-derived peptide inhibits retinal neovascularization and joint mobilization of bone cells...
... PEDF deriv peptid inhibit retin neovascular joint-mobilization bone-cell...
...PEDF deriv peptid inhibit aldehydes neovasular joint-mobilization bone-cells...
...Protein deriv peptid inhibit photoreceptor angiology manual-therapy cell-line...
...proteins deriv peptides inhibit vision medicine physical-therapy skeletal-system...
🟦 wiki
🟥 Entity
🟩 Entity&wiki
Input
B-CNN
Convolutional Neural Networks
convolution
k-max pooling
feature map
fully connection
yc
C
E
Entity feature
ye
Deep Boltzmann Machine
V
h1
h2
label
W1
W2
Wlab
q(h2/v)
Hierarchical classification
1
2
3
c
1
2
:
k
Output

Fig 1. CNNs-based hybrid learning framework.

https://doi.org/10.1371/journal.pone.0197933.g001

by convolution kernels and the networks can effectively detect the local features at every position of the preceding layer. The parameters for each nodes within the same feature map are the same for they share a same convolution kernel [30]. Due to this characteristics, the number of parameters within the network can be eliminated to a large extent compared with conventional neural networks, which makes the structure simpler and more flexible. Max pooling layer is utilized for feature mapping, and the operator is a nonlinear subsampling function that returns the maximum of a set of values [40].

In each convolution layer, every input sample vector is weighted with several different convolutional kernels, and various degrees of local features are extracted by varying the sizes of sliding window. Following the feature representation layers(convolutional and pooling layers), there are fully-connected layers where take the each feature maps as input and the neurons are connected like conventional networks. Through a certain number of convolutional kernels, the semantic representations of documents can be easily enriched.

Assume the input sequence matrix is A, the kernels used for convolution and pooling are K and β. And b_i^1, b_i^2 are respectively the bias vectors for convolutional layer and pooling layer. The output of convolutional layer, pooling layer and fully-connected layer are:

$$a_i^1 = f(conv2(A, K) + b_i^1) \tag{1}$$

$$a_i^2 = f(\beta down(a_i^1) + b_i^2) \tag{2}$$

$$y_c = f(wa_i^2 + b) \tag{3}$$

where *conv*2 is the convolution of two matrices, *down* represents the sampling operation to implement max-pooling, K, β, b_i^1, b_i^2 and the parameters w in fully-connected neural networks make up with the parameter set of CNN.

3.2 Deep Boltzmann machines

The DBM [31, 52], which is composed of RBMs, is an undirected graph network of symmetrically stochastic binary units. The RBM [53] is a generated neural network that can learn probability distribution over a set of inputs. More specifically, an RBM is a kind of Boltzmann machine in which all visible and hidden units are fully-connected between layers and not connected within each layer. The visible units ($\boldsymbol{V}$) represent the input of data, and the hidden units ($\boldsymbol{h}$) represent features learned from the visible units. The DBM has many advantages: it retains and discovers layer presentations of the input with an efficient pertaining procedure; it can be trained on unlabeled data, and parameters of all layers can be optimized jointly in the likelihood function. Fig 2 shows how the layer nodes are constructed in the DBM. This is a three-layer network with no within-layer connections. $\boldsymbol{h_1}$, $\boldsymbol{h_2}$ and $\boldsymbol{h_3}$ are three hidden layers with a different number of nodes for each level. The energy of the state $\{\boldsymbol{V}, \boldsymbol{h_1}, \boldsymbol{h_2}, \boldsymbol{h_3}\}$ is defined as:

$$E(V, h_1, h_2, h_3; \theta) = -V^T W_1 h_1 - h_1^T W_2 h_2 - h_2^T W_3 h_3, \tag{4}$$

where $\theta = \{\boldsymbol{W_1}, \boldsymbol{W_2}, \boldsymbol{W_3}\}$ are the model parameters, representing symmetric interaction terms.

During the process of calculating the value of the nodes in each layer, the undirected connection in each visible and hidden layers, which incorporates top-down feedback, plays a critical role in the performance of a DBM. This is the case for the DBM both as a generative and discriminative model. Additionally, this quality makes the feature representation of the DBM perform better than the deep belief network (DBN) [31].

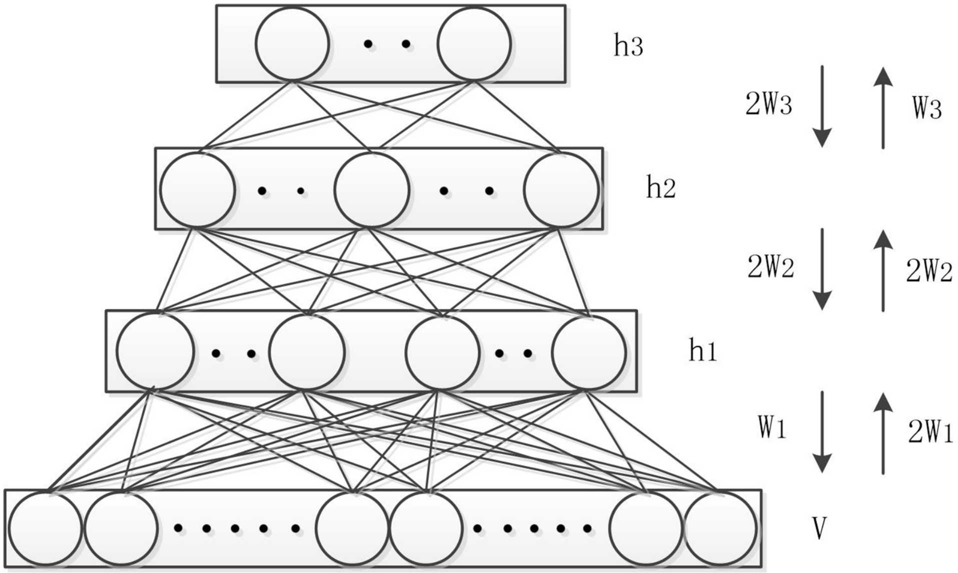

Fig 2. Deep Boltzmann machines.

https://doi.org/10.1371/journal.pone.0197933.g002

3.3 Boltzmann-Convolutional neural network (B-CNN)

3.3.1 Principle analysis. CNNs perform well when extracting local information about context. But for articles for which the abstract is the only available information, more global information may be required. Mikolov [54] showed that adding external context information can improve the performance of the model. The entity occupies an important position in the field of biomedicine indexing [55], so we combined the entity, as the complementary global feature information, with the input vector in each document.

The DBM, which updates weights using a priori knowledge learned from undirected graphs, represents a good feature extraction method in deep learning. Additionally, when analyzing the data structure of documents, we have found that the DBM can remove noise introduced by the document input. The DBM uses the nodes of both the former layer and the latter layer, as a result, hidden node sampling is more accurate. However, the DBM is disadvantageous in that the training time grows exponentially with the number of layers and nodes, as well as with the magnitude of the connection strengths [52]. To decrease the time complexity, we used a two-layer DBM, thus taking both training time and model accuracy into consideration for biomedicine indexing.

3.3.2 Framework analysis. In Fig 1, the entity feature, which represents the global features of each document, is a vector with the same dimensions as $\boldsymbol{y_c}$. The values ($\boldsymbol{V}$) in the input layer of the DBM are computed as follows:

$$V = f(\mathbf{C}y_c + \mathbf{E}y_e), \tag{5}$$

where

$$f(x) = \frac{1}{1 + e^{-x}}. \tag{6}$$

C, E are the weight matrices for which the likelihood of the training data is maximized. $\boldsymbol{y_c}$ is the output of the CNNs model, and $\boldsymbol{y_e}$ is the entity feature corresponding to each document.

$\boldsymbol{W_1}$, $\boldsymbol{W_2}$, $\boldsymbol{W_{lab}}$ represent weight connections between the visible layer and the $\boldsymbol{h_1}$ layer, $\boldsymbol{h_1}$ layer and $\boldsymbol{h_2}$ layer, and $\boldsymbol{h_2}$ layer and the label layer, respectively. We calculated the values ($\boldsymbol{V' - V}$) and ($\boldsymbol{h_1' - h_1}$), updating for weight. $\boldsymbol{V'}$ and $\boldsymbol{h'}$, which are the reconstruction representations after each sampling (corresponding to $\boldsymbol{V}$ and $\boldsymbol{h}$, respectively), represent the state of the visible and hidden nodes, respectively. The process of sampling the nodes in the $\boldsymbol{h_1}$ layer and $\boldsymbol{h_2}$ layer is shown in Fig 3,

$$h_1 \rightarrow h_2 : \ h_2 = h_1 * W_2 + tar * W_{lab}^T + bias1, \tag{7}$$

$$h_2 \rightarrow h_1' : \ h_1' = V * W_1 + h_2 * W_2^T + bias2, \tag{8}$$

$$h_1' \rightarrow h_2' : \ h_2' = h_1' * W_2 + tar * W_{lab}^T + bias3, \tag{9}$$

where *tar* represents the real label of the training sample, and *bias* is the bias value between the visible and hidden layers.

The node values in the $\boldsymbol{h_1}$ and $\boldsymbol{h_2}$ hidden layers are calculated according to the nodes in the previous and next layers. After the unsupervised pre-training process, we used a supervised fine-tuning process in which we introduced the labels of the samples. The purpose of the fine-tuning is to update the weights connecting visible (hidden) and hidden layers, and the objective function is to minimize the energy function in the model. The DBM is an undirected graph model, in which each neuron node in the same layer is computed according to the

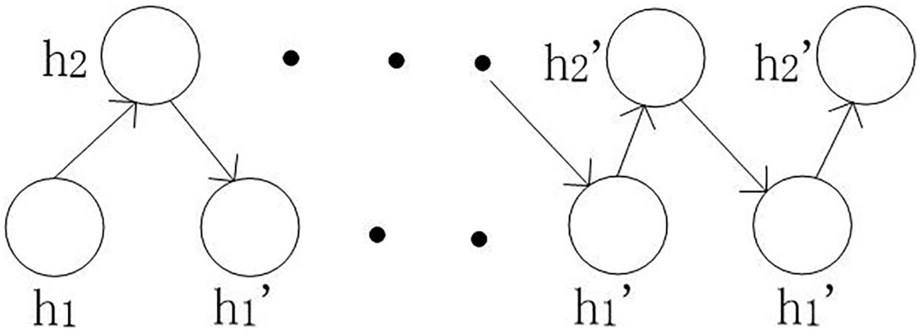

Fig 3. Layer's nodes sampling process.

https://doi.org/10.1371/journal.pone.0197933.g003

adjacent related lower and upper nodes. As the test sample labels are unknown, unlike the training sample labels, the training and testing processes of the DBM model are different. The details are described in the following section (Fig 4 is the training process and Fig 5 is the testing process)(Note that the $2W_1$ represents the $2W_1$ parameters illustrated in Fig 2 between layer V and layer h_1):

In the training process: During the training process, the model is sampled in the initial state and the nodes are reconstructed for each layer. After sampling the hidden layer nodes in the initial state ($\boldsymbol{h_1}$), which are calculated according to the visible layer nodes ($\boldsymbol{V}$), the subsequent calculation has two steps: calculate the status of the nodes in the second hidden layer ($\boldsymbol{h_2}$) (top row in Fig 4), and reconstruct the status of the nodes in the visible layer ($\boldsymbol{V}'$) (the lower row in Fig 4). Next, the labels of each sample are calculated, the status of $\boldsymbol{h}_1'$ and $\boldsymbol{h}_2'$ are determined

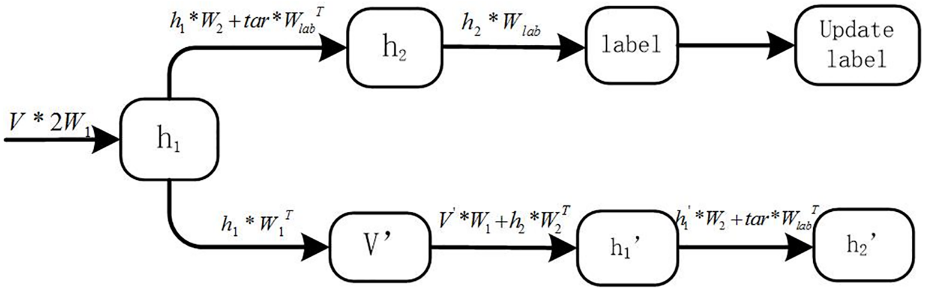

Fig 4. Training process.

https://doi.org/10.1371/journal.pone.0197933.g004

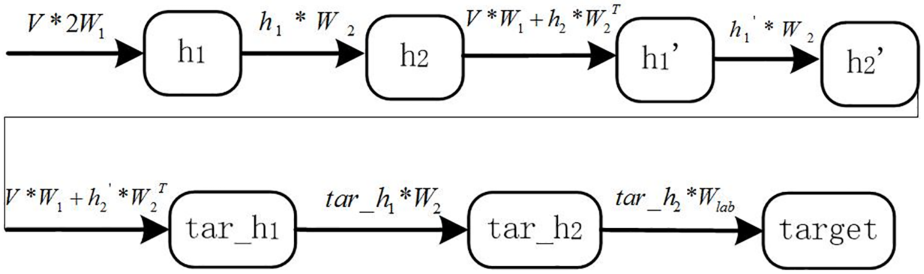

Fig 5. Testing process.

https://doi.org/10.1371/journal.pone.0197933.g005

based on the current status ($\boldsymbol{h_2}$), the error is calculated, and then the weights between the node connections are updated.

In the testing process: During the testing process, according to the DBM model node calculation rules (calculating the node status in the current layer according to the upper and lower layers), the weights obtained through the training process cannot be directly used to predict the test sample target. Indeed, the status of the $\boldsymbol{h_1}$ layer nodes (termed $\boldsymbol{h}_1^{'}$) must be re-sampled after calculating the $\boldsymbol{h_2}$ layer nodes. We sampled the status of the nodes in $\boldsymbol{h}_1^{'}$ according to the visual layer and the $\boldsymbol{h_2}$ hidden layer. We then re-calculated the $\boldsymbol{h_2}$ hidden layer nodes ($\boldsymbol{h}_2^{'}$) and predicted the labels of the test samples (details in Fig 5).

The rectified linear unit (ReLU) can be used to increase the nonlinear properties of the network, as well as the sparsity, without affecting the receptive fields of the convolution layer [56]. The ReLU function, which is also the neuron's output, is:

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

The sigmoid function, which is used widely in deep learning with excessive layers, suffers from the vanishing gradient problem; the training process is therefore difficult. Using the ReLU instead of the sigmoid function in a deep network ensures that the network neurons are modestly sparse after training, thus eliminating the issue of vanishing gradients along with the paths of active hidden units [16]. In addition, the ReLU enables faster network training compared with other options and does not require pre-training or advanced optimization strategies. Indeed, the ReLU can achieve performance comparable to a pre-trained model with the sigmoid function.

The recently introduced technique called "dropout" can reduce over-fitting by decreasing the complexity of co-adaptation of data [57]. Neurons are "dropped out" by randomly setting 50% of the nodes in each hidden layer in the network to 0, which means that 50% do not participate in the forward pass or back-propagation processes. In our system, "dropout" is used in all convolutional and fully-connected layers.

In the testing phase, dropout is regarded as a kind of mean network because for each input sample (which may be one sample or a batch of samples), the corresponding network structures are different but share the hidden node's weight at the same time. Hidden nodes appear randomly in a certain probability when updating weights in the network. One of the advantages of dropout is that it guarantees that every node does not appear at the same time. Hence, weight updating no longer depends on the interaction relationship between each pair of hidden nodes. This can prevent certain characteristics from affected by other specific characteristics so that each individual hidden unit can learn useful features without relying on other specific hidden units to correct its mistakes.

On the one hand, CNN has a great advantage on feature engineering. On the other hand, DBM shows great capability in generating the labels for documents while suffers from large number of parameters, which would lead to unbearable time cost. Compared with them, B-CNN is a combination of CNN and DBM, while the number of hidden layers in DBM can be reduced, replaced by CNN for extracting features from documents. As a result, in terms of time consumption, obviously B-CNN requires a larger computation cost than simple CNN or DBM with two hidden layers due to its complex pipeline. However, these methods cannot achieve the performances of B-CNN. Or if adding hidden layers to DBM, the time consumption cannot be controlled. Actually in practice, the feature engineering process, including CNN and meta feature extraction can be accomplished at the same time. With the optimized parameters, the testing process would not be influenced by the complex architecture that much.

3.4 Hierarchical indexing

Document classification aims at predicting the categories of an unknown document in a certain class collection. The task can be categorized into two groups through the number of target classes. Multi-class classification [58] is a common task in which there are obvious boundaries between each pair of classes and a given document can only belong to one class. Binary classification is a classical application of multi-class classification. Another task is multi-label classification [58], where each sample can be assigned with more than one class. The target labels can be either independent or associated which brings more challenging work for classifiers.

As a result, multi-label classifications are not simply repeated multi-class classifications due to the dependencies among labels. In fact, appropriate use of the dependencies can effectively improve the performances of classification. In case there are a large number of labels, through unsupervised clustering as pre-processing, the labels can be grouped into coarse subsets, which can greatly improve the efficiency of classification. Recently, word2vec [59] was developed to represent the word semantics from corpus through a continuous vectors, which has been found useful for obtaining the relationship between words by letting related or compositional words appear closed in the vector space. Based on this toolkit, Ioannis Pavlopoulos trained the word representation for biomedicine labels (http://participants-area.bioasq.org/), which is applied in our research. Based on the word vector representation, we design an indexing structure that introduces label (word) embedding for multi-label classification (shown in the third part of Fig 1). In our framework, the error function of the clustered coarse classes (categories) and the weight updating are used as follows.

$$E = \frac{1}{NC}\sum_{C}\sum_{N}\left(f(x, w) - y_c\right)^2, \qquad \Delta w = \frac{\partial E}{\partial w}, \tag{10}$$

where C is the number of coarse clustering categories, N is the number of training samples, x is the output label of B-CNN, w is weights which connect the output layer and the coarse cluster layer, $f(x, w)$ is the output probabilities of a certain label while y is the target value. For each C, the hierarchical classification function is:

$$J_k = \frac{1}{n_k}\sum\left(f(x, w_k) - y\right)^2, \qquad \Delta w_k = \frac{\partial J_k}{\partial w}, \tag{11}$$

where k is the number of categories within the corresponding coarse classes subset. The whole process is firstly clustering the labels into coarse classes with word2vec, and then deciding the final label within the coarse classes.

For forward propagation, through Eqs (4), (5) and (6), the output of the whole network can be computed with the input vectors and the parameters of CNN and DBM. Note that y_c can be represented by a function of input vectors and the parameters of CNN. For backward propagation, the parameters in CNN and DBM can be optimized simultaneously through stochastic gradient descent on the cost function.

4 Semantic feature representation

Currently, the main feature representation method is the vector space model with the BOW. However, the BOW model only contains information about word frequency, which is shallow and insufficient when dealing with NLP tasks, e.g., biomedicine indexing. In this paper, we describe a novel method for vector representation with document word sequence embedding

(DSE) for biomedicine documents, which is based on word sequence information. Our goal was to improve the feature representation (input of the framework). This model enables us to take advantage of the strength of the CNN, which is more suitable for dealing with the features of sequence relationships. We used the sequence of words in the documents as the basic features, and combined categorical information from Wikipedia with entity meta-features constructed from MetaMap keywords from biomedicine fields as a synonym for expansion. Afterwards, we used word embedding to represent these features, rendering this feature representation more informational for biomedicine tasks.

In the following, we first describe the meta-features composed via DSE representation, and then introduce the algorithm in detail. Wikipedia is a multi-language encyclopedia, covering a wide range of information on individual wiki pages. Wikipedia is an actual corpus, and with the rapid changes in social information, it is constantly being expanded and updated. Wu [60] exploited Wikipedia as a source of external knowledge for extending documents. Considering the nature of our task with biomedical abstracts, we adopted this idea and used Wikipedia to extend the documents. Named entity occupies an important position in addressing NLP tasks, e.g., information extraction and information retrieval. Named entity is a representative feature in biomedicine [61], e.g., the prediction of gene sequences and identification of proteins. Word vectors [59] are a good tool for the representation and mining of existing semantic relationships between words. Among existing deep learning methods, many models that obtain good performance are based on word vectors. Thus, for the input of the model, we used word vectors to represent Wikipedia and the named entity features.

4.1 Meta-features

4.1.1 Wikipedia as meta-feature enrich document representation. For the task of indexing biomedical documents, we enriched the document representation by using Wikipedia category information together with MetaMap keywords. This enabled us to overcome the shortcomings of the BOW. MetaMap (http://metamap.nlm.nih.gov/) is a widely used open source toolkit that extracts concepts in the UMLS metathesaurus. We computed the distribution of categories-words according to the known distribution of word-documents and category-documents using a process referred to as LDA (see Fig 6). This is discussed later in this document.

Parameter description: n is the number of distinct words in the documents, m is the total number of classes in all documents, and t is the number of Wikipedia categories. Matrix $\mathbf{P_1}$ represents the distribution of words and documents, where the row $\boldsymbol{w_i}$ is the probability that the word w_i will appear in m classes of documents, denoted as w_{ij}. Matrix $\mathbf{P_2}$ represents the distribution of words and Wikipedia categories, where the row $\boldsymbol{w_i}$ is the probability that a word w_i will correspond to a Wikipedia category, denoted as w_{ik}. Matrix $\mathbf{P_3}$ represents the distribution

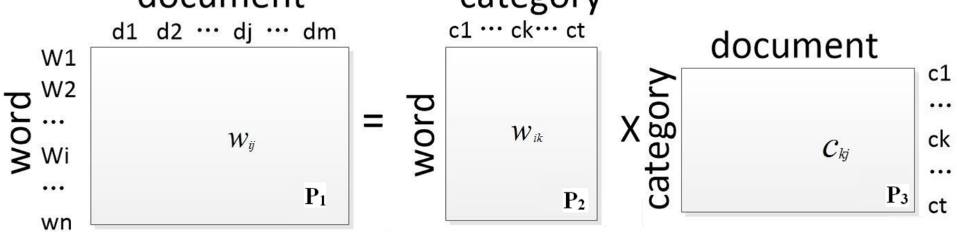

Fig 6. Words mapping with Wikipedia.

https://doi.org/10.1371/journal.pone.0197933.g006

of Wikipedia categories and document classes, where the row $\boldsymbol{c_k}$ is the probability that the category c_k will appear in the documents, denoted as c_{kj}.

We were able to use the Wikipedia categories for the corresponding anchor words to calculate the Wikipedia categories of non-anchor words ($\mathbf{P_2}$). This form of document representation combines global (the proportion of the words in all documents categories) and local (anchor words in each document category) information. Hence, through the text representation, we were able to learn more semantic information via the model.

4.1.2 Entity as meta-feature enriches document representation. Conditional random fields (CRFs) are undirected statistical graphical models that have been used successfully in a large number of studies on named entity recognition. CRFs take advantage of sequence labeling, which is a linear chain that corresponds with conditional training to treat each sentence as a sequence of tokens. In this paper, we used the Machine Learning for Language Toolkit (http://mallet.cs.umass.edu/) (MALLET), as an implementation of CRFs [62], to recognize entity class in biomedicine abstracts. We used biomedicine entities to enrich document representation in two different ways. First, we used the entity properties in the DSE, and second, the entities were included in the DBM inputs as an entity feature vector.

4.1.3 Word vector. The word vector model can be seen as a language model that consists of n-dimensional continuous valued vectors, where each dimension of the word embeddings expresses a latent feature of the words, exposing useful semantic and syntactic regularities [59]. The vector creates features without human intervention. Based on individual words and random initialization, it can make highly accurate guesses about the meaning of each word with enough data, usage, and context. These features are also called the distributed representation of the words.

Mikolov trained word representations using an effective neural network model that seeks to maximize accuracy while minimizing computational complexity [63]. The representation of vectors after training not only positions similar words close to each other in the vector space, but can also help to define implicit relationships among words across a distance. For example, the word vector ("King") – vector ("Man") + vector ("Woman") can have specific properties such that it results in the vector that is closest to the vector representation of the word Queen [64].

4.2 Document word sequence embedding (DSE)

We preprocessed all documents by removing stop-words and stemming. To fix the length of all document input features, we generated the following process. The document is represented as follows: if the document length after preprocessing is less than the average length L, we use the *Unknown* word as a supplement until the document length reaches L. If the document length is longer than L, we reprocess the document and retrieve high frequency words until the length of the document matches that of L, and then use word embedding according to the sequence of the words in the document. This process was termed algorithm 1.

We sorted all words in a document k according to their frequency in the document. We then collected the top words in the document until the document length equaled L. $\boldsymbol{D}$ represents documents after the process of removing stop-words and stemming, $\boldsymbol{D_k}$ represents document k, and *Unknown* refers to the word *Unknown*, which is used for effective word embedding in a large corpus. When the document length is less than L, we use this word to supplement the length. $L - m$ in $Unknown_{L-m}$ refers to the number of *Unknown* words. If the document had only one *Unknown* word, we just used $Unknown_1$. $\boldsymbol{R_{km}}$ refers to the word embedding of the m word in document k, and $\boldsymbol{R_{newkm}}$ describes m word embedding in the sequence of new words after the document is rearranged.

Algorithm 1: Document word sequence embedding (DSE)

```
Input: $D$
Output: document representation $R$
$L$ = average length of $D$
for $k$ = 1..$N$ do
    $m$ = length ($D_k$)
    if $m < L$
        $R = [R_{k1}, R_{k2}, ..., R_{km}, Unknown_1, ..., Unknown_{L-m}]$;
    else if $m = L$
        $R = [R_{k1}, R_{k2}, R_{k3}, ..., R_{km}]$;
    else if $m > L$
        sort all words in document $k$ according to their
        frequency in this document. Then take the top words
        in all place of this document until the length equal
        to $L$. $R = [R_{newk1}, R_{newk2}, ..., R_{newkm}, ..., R_{newkL}]$;
    end if
end for
Return $R$
```

In addition, we use the entity and Wikipedia categories for the corresponding anchor words to enrich document representation [65]. These were based on MetaMap keywords in the dataset of the biomedicine literature. Some words in the document may be both biomedical entities and also anchor words in Wikipedia. In these cases, there may be substantial corresponding category information. Taking into account the integrity of the model input, we selected three words to enrich document representation for each word in the ***R***. These words were selected from the corresponding categories and entities. The extension of word order followed that of the original words. Then, word embedding was used to represent each word. When the number of categories was less than three, we used the word itself in the document, together with its categories. Each sentence in the document was thus represented by word embedding, in which each word was represented by one 50-dimensional vector. As a consequence, each document was represented by a 200 (50×4)×L matrix. The details of the process are shown in the feature representation section of Fig 1.

5 Experiments

5.1 Dataset and experimental setup

Dataset 1 (http://tinyurl.com/m2c8se6) was a labeled corpus of English scientific medical abstracts. All data were downloaded from the Springer website. Dataset 2 (http://biotext.berkeley.edu/data.html) was a collection of BioText data. Dataset 3 (http://www.bioasq.org/participate/challenges) was a group of MEDLINE documents. For dataset 3, we downloaded all data for the years up to and including 2013 from the website. Each article contained only the article title and abstract, and there were 5 to 20 classes (termed Medical Subject Headings in the medical field (MeSH)) for each article on average. Altogether, there were 27149 MeSH headings. Yepes [27] selected the top 10 most frequent MeSH headings to address the extremely unbalanced distribution of the dataset. Similarly, we selected the top 150 MeSH headings because only about 150 of these appear in more than 1% of all the data on MEDLINE. To extend our experiments, we plan to introduce some unbalanced samples and select the top 2000 categories. Dataset 4 (http://qwone.com/~jason/20Newsgroups) was organized into different newsgroups, each corresponding to a different topic. The website offered three versions of this dataset, and we selected the third, which had 18828 documents. This dataset can be used in two ways, as it can be organized into twenty classes or four major categories (comp, politics, rec, and religion). To ensure that our results were comparable to those of Lai [30], we

Table 2. Datasets details.

Dataset	Sample	Classes	Type	Field
dataset 1	9666	39	multi-class	biomedicine
dataset 2	1000	168	multi-class	
dataset 3	1,000,000	150	multi-label	
		2,000	multi-label	
dataset 4	18,828	20	multi-class	newsgroups
		4	multi-class	

https://doi.org/10.1371/journal.pone.0197933.t002

choose to use the same four categories of the dataset 4. For more information about the datasets, see Table 2.

On dataset 1 and 2, we aim to primarily compare the performance of different models and different feature representations. To simplify the complexity, on dataset 3, we choose not to conduct the experiment on the BOW_+ feature representation because of the unsatisfying performance, compared with DSE. Dataset 4 is used to verify the extensions of the proposed model on fields other than biomedicine, so we directly compare our model with state-of-the-art models in the chosen field, rather than the baselines of the former datasets.

These experiments are repeated ten times. In order to avoid insufficient testing set to validate the effectiveness, as well as to guarantee the training set sizes, at each time, 70% of the entire dataset is randomly selected as training samples, and the remaining 30% as testing samples. The presented results are the average of results of the repeated experiments. Moreover, for each, within the selected training samples, 10-fold cross validation is employed to select the optimal parameters. The parameters of the B-CNN model include the settings of the convolutional and pooling layers in CNN, the size of the sliding window in CNN, the interlayer connection weights (e.g. $\boldsymbol{W_1}$, $\boldsymbol{W_2}$, and $\boldsymbol{W_{lab}}$ in DBM, the connection parameters of weights from the high-level representation layer in B-CNN, and the initial values in the pre-training process and the overall learning rate, and so on. We applied the widely-used cuda-convnet package to train our model on a single GPU. The experimental performance presented below was conducted entirely with the test set.

5.2 Details

Based on our DSE feature extension, we also enriched BOW representation through Wikipedia and the entity class based on MetaMap. We used this as our baseline, denoted as BOW_+ (BOW_+ is better than BOW representation. BOW_+ adopts indeed the same representation approaches of conventional BOW). In addition, we made some changes when training CNNs. As our purpose was to learn about the structure of the document and the semantic information between words, we fixed the size of the slide window to make it equal in dimension to that of word embedding (50-dimension). We used a step down sliding window position of 50 by 50, as this prevented each sliding window from changing the word vector embedding associated with the features of the word. In this way, information about the document was gathered by changing the length of sliding window.

5.2.1 Evaluation protocols. We used Macro- and Micro-averages of the precision (P), recall (R), similarity (S), and F_1-measures as the evaluation criteria [66–69]. The Macro-average weights all the classes equally, regardless of how many documents are included. The Micro-average weights all the documents equally, thus favoring the performance of common classes. Hence, the Macro-average reflects the performance of each category, and the Micro-average reflects the performance of each document. MacroP (MaP), MacroR (MaR), MacroF_1

(MaF_1), MicroP (MiP), MicroR (MiR), and MicroF_1 (MiF_1) are comprehensive assessment metrics, defined as:

$$MacroP = \frac{\sum_{k=1}^{K} P_k}{K}, \tag{12}$$

$$MacroR = \frac{\sum_{k=1}^{K} R_k}{K}, \tag{13}$$

$$MacroF_1 = \frac{\sum_{k=1}^{K} F_{1k}}{K}, \tag{14}$$

$$MicroP = \frac{\sum_{k=1}^{K} TP_k}{\sum_{k=1}^{K} TP_k + \sum_{k=1}^{K} FP_k}, \tag{15}$$

$$MicroR = \frac{\sum_{k=1}^{K} TP_k}{\sum_{k=1}^{K} TP_k + \sum_{k=1}^{K} FN_k}, \tag{16}$$

$$MicroF_1 = \frac{MicroP \times MicroR \times 2}{MicroP + MicroR}, \tag{17}$$

where K is the number of categories. Precision (P) is defined as: $P = \frac{TP}{(TP+FP)}$ and recall (R) as: $R = \frac{TP}{TP+FN}$. F_1, Macro-similarity (MaS) and Micro-similarity (MiS) are defined as:

$$F_1 = \frac{2 \times P \times R}{(P + R)}, \tag{18}$$

$$MaS = MiS = \frac{TP}{(TP + FP + FN)}, \tag{19}$$

where TP refers to the number of true positives, FP is the number of false positives, FN is the number of false negatives, and we also use TN which is the number of true negatives in the Roc graph.

A ROC [70] graph depicts the relative tradeoffs between the benefits (TP, true positives) and costs (FP, false positives). ROC curves are two-dimensional graphs in which the TPR (true positive rate) is plotted on the Y axis and the FPR (false positive rate) is plotted on the X axis. In the ROC curve, the closer the point to the top left, the better the model. The TPR and the FPR of a classifier are estimated as:

$$TPR = \frac{Positive\ correctly\ classified}{Total\ positives} = \frac{TP}{(TP + FN)}, \tag{20}$$

$$FPR = \frac{Negatives\ incorrectly\ classified}{Total\ negatives} = \frac{FP}{(FP + TN)}. \tag{21}$$

5.2.2 Comparative methods. We compared the performance of our best model (with five pooling layers, each of which follows a convolutional layer, and three fully-connected layers at the end of CNNs) with MTI, MeSHUP, which we have described in our related work, and typical methods, as follows.

Pattern Matching (PM) [19]: By comparing MeSH labels with words in the documents, we were able to predict document labels. During the matching process, we added some artificial rules: convert all words in the MeSH label into lowercase. In addition, we determined the characters in MeSH labels using fuzzy matching. For example, it might be unlikely to see *pain bone* in a MeSH label, but very likely to find *pain in bone* in a document.

Latent Dirichlet Allocation (LDA) [33]: The LDA is a directed graphical model, also referred to as the topic model. This model is an unsupervised learning technique for extracting thematic information from a corpus. LDA, HLDA (Hierarchical LDA), and PLSA (probabilistic latent semantic analysis) have been extensively used for classification because the nonparametric extensions of these models have been quite effective [71]. The main purpose of LDA is to reduce the dimensions of document representation using thematic dimension representations instead of the dictionary. This motivation of the LDA method is similar to that of deep learning, in which features are extracted from high-dimensional space. Variational EM algorithms and Gibbs sampling are used in the LDA model, which makes two of the parameters, i.e., document-topics and topic-words, easier to predict.

Support Vector Machines (SVM) [72]: In machine learning and data mining, the SVM is supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. In this paper, the multi-label classification produced by SVMs is regarded as a group of multiple binary classification problems. The samples from each class in the dataset have an unbalanced distribution. For each class, positive samples are defined as current class samples, and negative samples are two to three times the size of the positive samples. As described in the reference [73], when the number of features is small, one often maps data to higher dimensional spaces (i.e., using nonlinear kernels). Considering the datasets in the experiments, the RBF (Radial Basis Function) kernel is selected and used which has two parameters, C and γ. In SVM, the RBF kernel is selected by considering two factors: the first reason is that this kernel nonlinearly maps samples into a higher dimensional space; correspondingly the second reason is the number of hyper-parameters that influences the complexity of our model. In the experiments, cross-validation was applied to select the best pair of (C, γ) from various pairs of candidate parameters. In addition, we also tried the linear kernel (LIBNEAR) to compare with LIBSVM with the RBF kernel.

Naive Bayesian (NB) [74]: The NB is a supervised probabilistic learning method. The basic idea of this method involves the calculation of appearance probability under some class, returning the label of the maximum probability.

Logistic Regression (LR) [74]: The most simple form of regression is linear regression. In LR, a logic function is applied on the basis of linear regression. In LR, a set of weights is learned by training with classifiers. In the test phase, the set of weights can discriminate results according to training samples. Unlike LR, which uses logistical loss as the objective function, the SVM uses the hinge loss function. The SVM method considers support vectors, which are the most relevant points for defining the classification boundary when training the classifiers. Alternatively, the LR method uses nonlinear regression mapping, which greatly reduces the weight of points that are far away from the classification boundary, and relatively improves the weight of the most relevant points.

Hierarchical CNN (HC): In HC, the feature representation of a document is learned by CNN training only, and then hierarchical classification is conducted.

CNN+DBN: We verified the validity and necessity of the DBM component of our B-CNN model by using DBN instead of DBM for biomedicine indexing. DBN is also a good feature extraction method in deep learning, and like DBM, is composed of stacked RBMs. However, the two differ in that the DBN is a directed graph model, that calculates the hidden layer nodes based on the previous layer nodes only, while the DBM is an undirected graph model, in which the state of hidden layer nodes is decided jointly by the upper and lower layers (the detailed calculation process is shown in Figs 4 and 5). To compare the DBM and DBN in this paper, we selected two-layer RBMs for feature representation.

Directly Binary Classification (DBC): In our experiments, in order to verify the validity of the hierarchical classification in our model, we compared our method to flat classification based on the CNN model, called DBC. For each class, a binary classification can be utilized to extend each node into two nodes. The labels are modified for each document as follows: if they are labeled (1, 0), it means the document belongs to this class, and if they are labeled (0, 1), it does not belong to the class. The output values of the nodes are computed by the sigmoid function. Comparing the values of the two nodes determines whether the document belongs to a specific class.

5.3 Experimental results

In this section, we analyzed the model performance using 11 models with four evaluation metrics, four datasets, and two feature representations. We conducted experiments in three parts. In the first part, we compared and analyzed biomedicine indexing performance with different feature representations using dataset 1 and dataset 2. In the second part, we analyzed the effects of the different models in terms of features representations using dataset 3. ROC analysis and significance tests are described in the third part.

We present the experimental results for the 11 methods based on the DSE and BOW_+ features in the tables listed in the experimental results section for dataset 1 and dataset 2 (horizontal comparative analysis of the models with different features, comparative analysis of the longitudinal effects of different models with the same features). The figures show the experimental results for dataset 3. As dataset 4 was not from a biomedical field (we wanted to verify the extensions of the B-CNN model), these data are analyzed separately in section 5.4.

5.3.1 Experiment with different feature representations. When comparing BOW_+ in terms of DSE features, each document is represented as a matrix based on the DSE feature or as a vector based on the BOW_+ feature. The SVM, LDA, LR, and NB methods involve learning global features, so the input features (DSE) of the document are read by the columns of the matrix during training; that is, each document is represented as a vector of $200{\times}L$.

As can be seen in Tables 3, 4 and 5, our B-CNN method with DSE feature representation had the best performance among the models. However, for the shallow learning (except HC, CNN+DBN, and DBC) models, the DSE was not superior to the BOW_+. This is likely because these models learn using global features, and different documents in the same dimension can be represented by the same word in BOW feature representation. Conversely, the DSE cannot distinguish between different words in these shallow learning models, and so there is an increase in noise.

While the DSE is not suitable for these shallow learning models, it does show good performance in the CNN model. There are several reasons for this. First, this representation considers the words sequence of the document. As mentioned above, CNNs are more suitable for dealing with features with a sequence relationship, which is also the reason behind their contribution to speech research [14, 51]. Although the words in the same dimension are not fixed, each word has a 50-dimensional word embedding representation that is easily distinguished from that of other words. Indeed, setting the size of the sliding window and learning the local information between words can suppress the emergence of noise. The noise comes mainly from the stochastic gradient descent procedure. It can be attributed to the approximate sampling procedure what cause systematically biased estimates of the gradient [75]. DBM can be trained in a semi-supervised manner with labels connected to the top layer. It can improve

Table 3. Classification results (%) on dataset 1.

Method	BOW_+						DSE					
	MiP	MiR	MiF_1	MaP	MaR	MaF_1	MiP	MiR	MiF_1	MaP	MaR	MaF_1
PM	21.90	31.56	25.86	22.12	39.45	28.35	21.90	31.56	25.86	22.12	39.45	28.35
LDA	55.15	54.77	54.96	56.28	55.89	56.08	50.98	50.98	50.98	51.49	50.98	51.23
SVM	57.44	57.04	57.24	58.02	57.61	57.82	47.43	47.00	47.21	47.05	46.63	46.84
NB	53.47	55.86	54.64	55.70	55.31	55.50	50.53	49.93	50.23	50.64	50.43	50.53
LR	54.98	53.54	54.25	54.54	55.19	54.86	50.56	46.52	48.46	50.97	46.89	48.85
MTI	62.03	**63.86**	62.93	62.66	62.61	62.63	52.27	52.11	52.19	52.22	52.06	52.14
MeshUP	63.95	59.62	61.71	62.69	62.10	62.40	50.66	50.20	50.43	50.56	50.10	50.33
HC	60.65	58.36	59.48	61.89	59.55	60.70	63.20	61.05	62.11	66.93	63.75	65.30
CNN+DBN	63.56	61.93	62.73	63.00	61.80	62.40	65.98	64.36	65.16	68.19	64.94	66.52
DBC	54.84	52.39	53.59	55.12	52.91	53.99	58.96	57.56	58.25	59.08	58.26	58.67
B-CNN	**64.96**	63.30	**64.12**	**64.54**	**62.87**	**63.69**	**68.51**	**66.89**	**67.69**	**70.31**	**66.20**	**68.19**

https://doi.org/10.1371/journal.pone.0197933.t003

performance and simplify the training of deep generative models. It also controls noise and enables training via backpropagation like standard deep supervised networks [76]. Finally, CNNs are trained by convolution, pooling, and ReLU, and these nonlinear layers also represent a good solution for the information interference problem introduced by using *Unknown* word when the document length is less than *L*.

Second, the BOW_+ representation is too sparse, especially in the case of unbalanced data, where the difference between the feature representations of documents is small. Categories with a large number of samples greatly interfere with categories with fewer samples, leading to predictions that are generally biased to high frequency labels.

Summarily, a comparison of our experimental results for DSE and BOW_+ features shows that the DSE (based on word embedding) clearly retrieves more semantic knowledge and extracts a more accurate representation of the document.

5.3.2 Experiment with different models. The HC model, in which the entity characteristics do not enrich the document, has high precision but low recall. Upon further examination of the labels of the classification results, we found that most of the missing labels classified by CNNs are MeSH terms. Few samples belong to these tags in the datasets, so it is not easy for CNNs to capture these features. However, the advantages gained by combining the CNN with

Table 4. Classification results (%) on dataset 2.

Method	BOW_+						DSE					
	MiP	MiR	MiF_1	MaP	MaR	MaF_1	MiP	MiR	MiF_1	MaP	MaR	MaF_1
PM	25.57	36.78	30.17	25.27	36.78	30.17	25.57	36.78	30.17	25.27	36.78	30.17
LDA	47.27	46.94	47.11	47.04	46.00	46.52	40.06	39.66	39.86	40.10	39.70	39.90
SVM	48.74	49.71	49.22	48.79	49.76	49.27	40.96	40.59	40.77	40.87	40.14	40.50
NB	46.79	46.46	46.62	45.71	44.97	45.34	40.68	40.28	40.48	39.46	38.67	39.06
LR	45.81	45.90	45.86	45.03	45.12	45.08	38.11	35.06	36.63	38.30	36.05	37.14
MTI	52.01	51.96	51.99	49.93	50.92	50.42	42.28	42.16	42.22	44.52	41.31	42.86
MeshUP	51.41	52.90	52.14	51.51	**53.06**	52.27	39.54	39.19	39.37	39.47	40.09	39.78
HC	52.00	50.51	51.25	51.39	47.32	49.27	56.01	54.23	55.11	53.21	50.43	51.78
CNN+DBN	54.70	52.57	53.61	53.85	50.07	51.89	58.60	56.19	57.37	55.87	52.50	54.13
DBC	47.30	43.51	45.32	46.96	43.08	44.94	49.50	50.50	49.99	48.95	48.93	48.94
B-CNN	**56.52**	**54.46**	**55.47**	**55.97**	52.51	**54.18**	**60.94**	**58.49**	**59.69**	**58.92**	**55.94**	**57.39**

https://doi.org/10.1371/journal.pone.0197933.t004

Table 5. Similarity measure (%) on dataset 1 and dataset 2.

Method	dataset 1				dataset 2			
	BOW_+		DSE		*BOW_+*		DSE	
	MiS	Mas	MiS	MaS	MiS	MaS	MiS	MaS
PM	40.56	41.94	40.56	41.94	42.08	42.08	42.08	42.08
LDA	52.61	53.24	50.49	50.62	48.60	48.32	45.40	45.41
SVM	53.90	54.24	48.64	48.47	49.62	49.64	45.78	45.66
NB	52.45	52.91	50.12	50.27	48.37	47.77	45.65	45.07
LR	52.22	52.56	49.28	49.47	48.01	47.65	44.09	44.32
MTI	57.44	57.23	51.12	51.09	51.01	50.21	46.39	46.69
MeshUP	56.68	57.08	50.22	50.17	51.10	51.17	45.19	45.36
HC	55.25	56.00	56.90	59.06	50.64	49.68	52.70	50.93
CNN+DBN	57.30	57.08	58.94	59.93	51.89	51.00	53.99	52.18
DBC	51.87	52.09	54.50	54.75	47.80	47.63	50.00	49.48
B-CNN	**58.23**	**57.94**	**60.75**	**61.17**	**52.90**	**52.21**	**55.38**	**54.02**

https://doi.org/10.1371/journal.pone.0197933.t005

the entity features retrieved by the DBM compensate for the tendency of the CNN to focus on local information.

We compared the CNN+DBN model with a version of the B-CNN model in which the DBM had been replaced by the DBN. From Figs 7, 8, and 9, we can see that the CNN+DBN

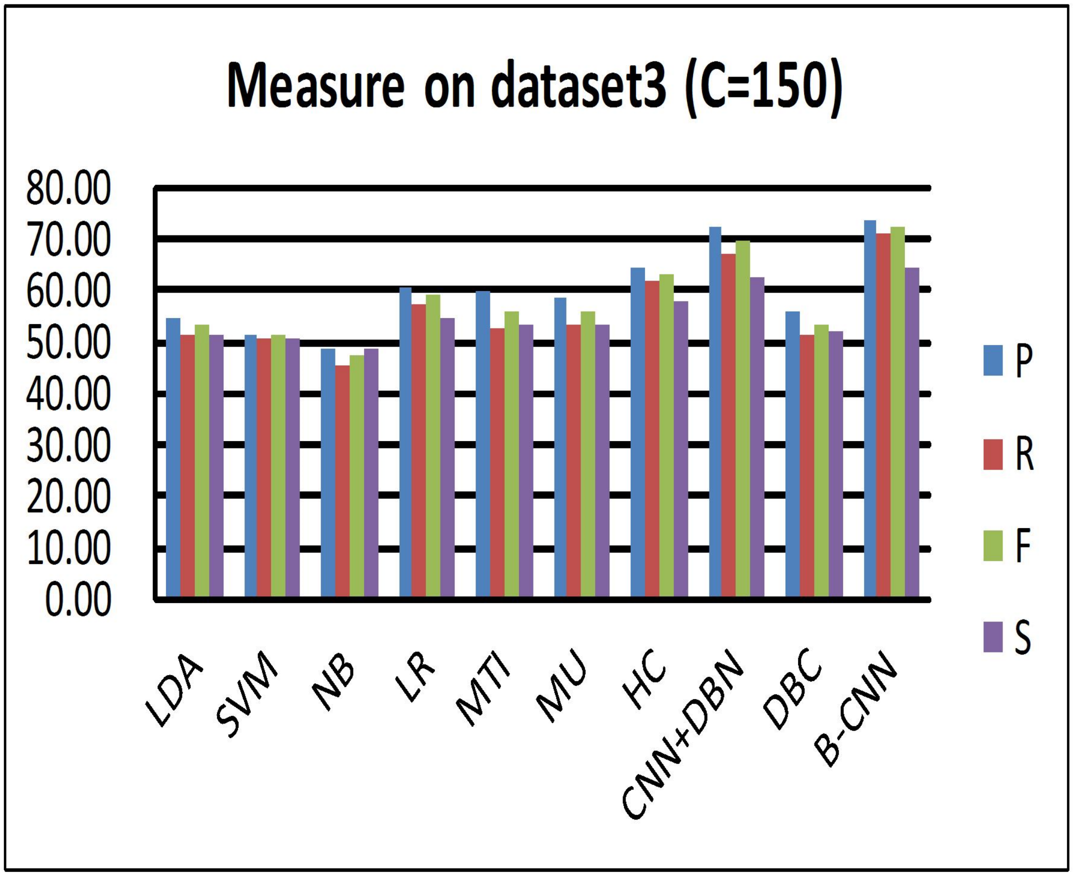

Fig 7. Measures (Precision recall F_1 similarity) on dataset 3 (C = 150).

https://doi.org/10.1371/journal.pone.0197933.g007

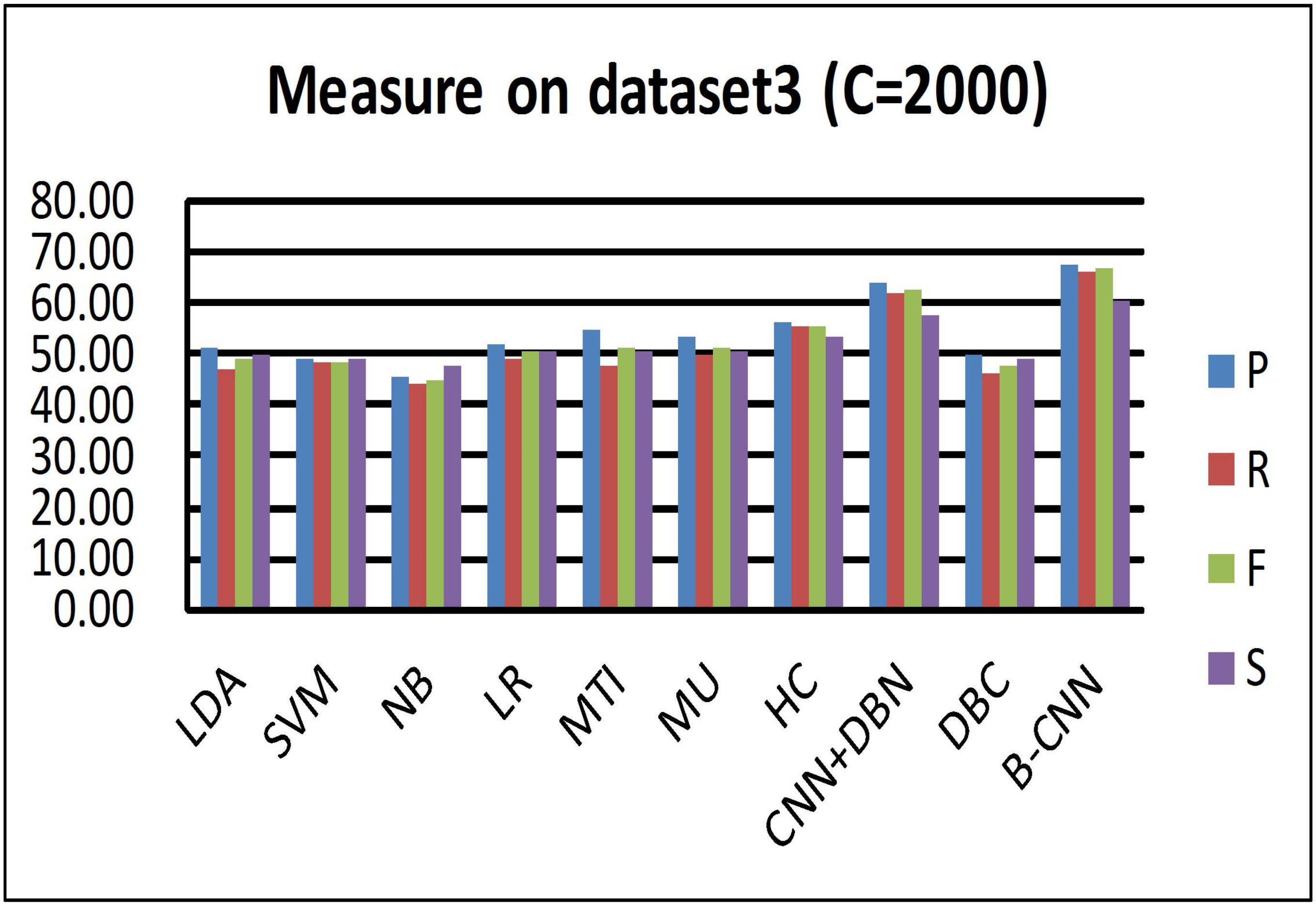

Fig 8. Measures (Precision recall F_1 similarity) on dataset 3 (C = 2000).

https://doi.org/10.1371/journal.pone.0197933.g008

model is better than the HC model but still not as good as the B-CNN model. This undirected graph model can adjust the weights that connect the nodes between visible and hidden neurons, and it is thus more flexible. To further explore why the B-CNN performs better than the CNN+DBN model, we checked the document representation in the high-level components of the DBM and DBN. We found that the cosine value calculated in the DBN model was larger than that in the B-CNN model. Generally, a larger cosine value indicates that the two samples are more similar. Although the input vector of the document was the same, the representation of the document differed between the two models. The smaller cosine value for the B-CNN model indicates that the test document representation of the model output node dimensions was more refined. This also means that the B-CNN model can learn more deeply distributed representations from documents between different samples, which can help the new test document retrieve more similar samples from the training samples.

For the DBC model, Tables 3 and 4 and Fig 9 (this figure shows the performance of hierarchical classification on the C = 2000 dataset, which has a similar performance on C = 150) all show that hierarchical indexing performed much better than flat classification (DBC) when processing a large number of classes. Moreover, the coarse cluster step is an effective way to remove noise from the unbalanced distributed samples. By analyzing the results from the test documents, we found that most test samples were predicted to be negative by the model.

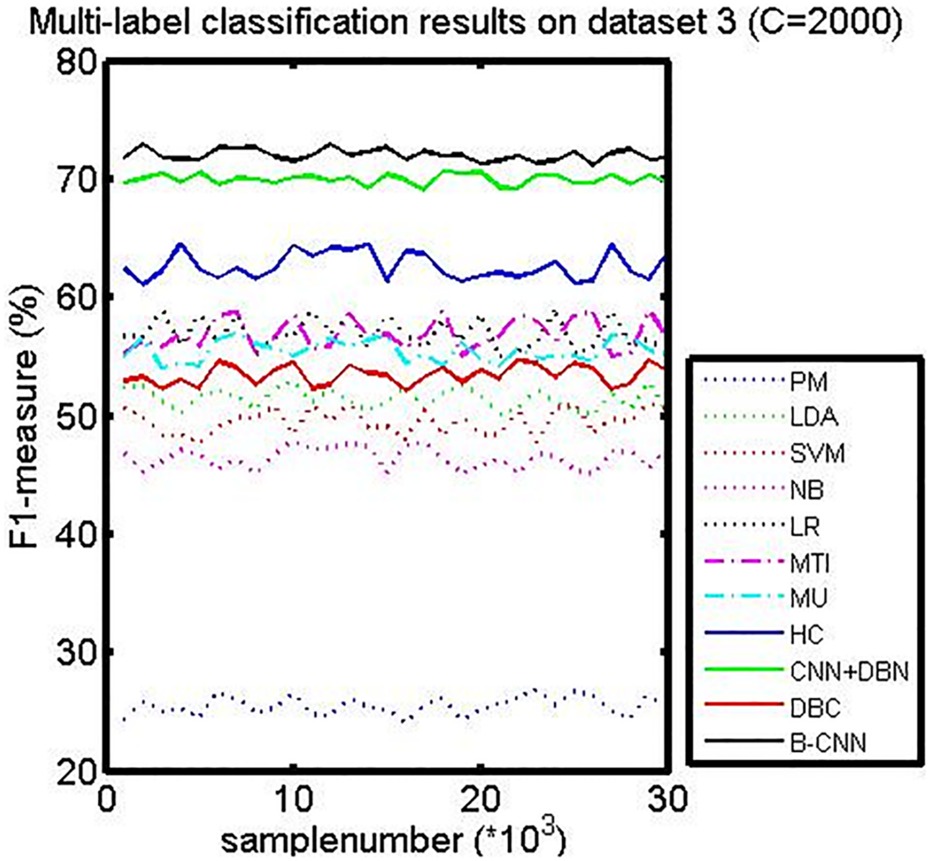

Fig 9. F_1-measure curves on dataset 3.

https://doi.org/10.1371/journal.pone.0197933.g009

Further evaluation of the training stage revealed very few class numbers of positive samples. Too many nodes were connected in the model, and the adjustment of their weights was not updated in the fine-tuning process. Specifically, more weights were updated in the negative samples compared with the positive samples.

Our proposed hierarchical semantic indexing method (B-CNN) greatly increased the classification precision for the positive samples in the first layer. Fig 10 is the label embedding representation of coarse clusters. This figure shows the classification results of the samples, which were previously colored based on the actual classes of the chosen samples. The distances between each two clusters proves the effectiveness of clustering of the proposed model. As shown in the figure, through the coarse classification process, the samples are grouped into 10 obvious clusters. The colors of the samples are based on the co-occurrence of the words. That is, if two labels contain some similar words, they may be in the same or similar colors. The phenomenon whereby the points with the same color are not grouped together indicates the fact that the co-occurrence of the words cannot completely represent the semantics. In summary, the distance between each pair of clusters proves the effectiveness of coarse classification, while the distribution of the same colors indicates that the relations among labels cannot be represented by co-occurrence. Hence, our future research work should consider other relations among the labels to better construct a hierarchical classification tree. This hierarchical semantic framework can effectively reduce the negative impact of negative samples, and mainly updates the weights connected with previous layer nodes during the next layer classification.

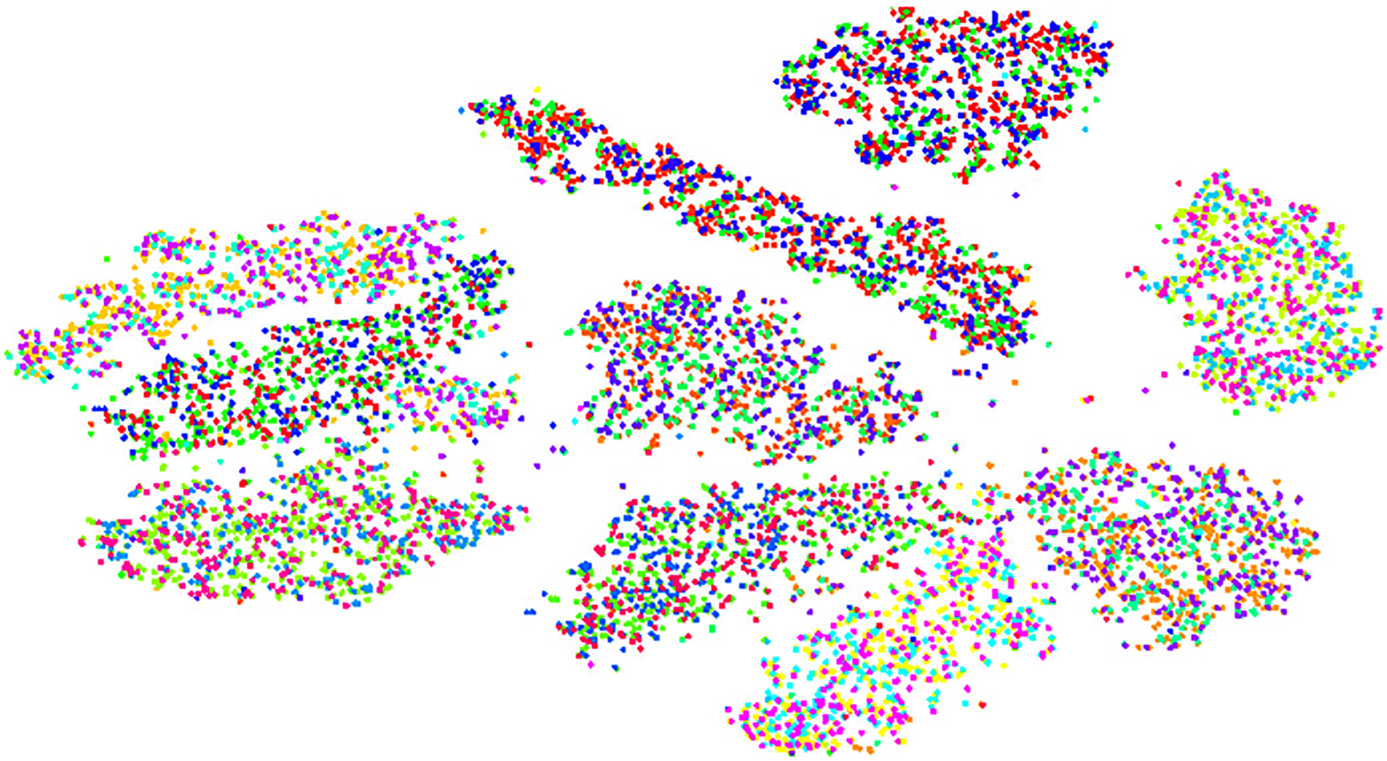

Fig 10. Coarse clusters of label embedding.

https://doi.org/10.1371/journal.pone.0197933.g010

The updates to weights of nodes that are connected with different previous layer nodes are independent from one other (see Eq 11), which greatly improves the efficacy of this model for training of positive samples. We also compared the ReLU with the sigmoid function in our B-CNN model. The network neurons with ReLU were reasonably sparse after training.

To summarize, the efficacy of our approach owes not only to the hierarchical indexing architecture, but also the feature representation. MetaMap can extract concepts that appear in the UMLS from biomedical texts. These vocabularies can provide better representations for retrieving relevant MEDLINE citations.

5.3.3 ROC analysis. The four ROC curves (shown in Fig 11) for the three biomedical datasets were produced by six different methods (the SVM and LDA are state-of-the-art classification methods in shallow learning, and the others are deep learning methods). The reasons to select the ROC measure to analyze the performances are mainly derived from the effectiveness (compared with precision, recall, F_1-measure, and similarity) and the robustness of different models. These two performances can be effectively evaluated through the trend of the ROC curves of different models. Obviously, the best curve trend is continuously rising. To better analyze and compare the performance of different methods, we amplified and focused on the results of the ROC partial curve (the abscissa range only shows the data values between 0 and 0.5). We followed the papers [77, 78] and downloaded the codes from (http://www.mathworks.com/matlabcentral/profile/authors/1336198-stefan-schroedl).

Fig 11 shows the ROC graphs, from which the performances of different models can be apparently distinguished. In these figures, the closer to the upper left the curve is, the better the performance of the corresponding model is. These curves shows our analysis of the effects of different models on the datasets: (1) all of the curves in the figure are accompanied by a slight

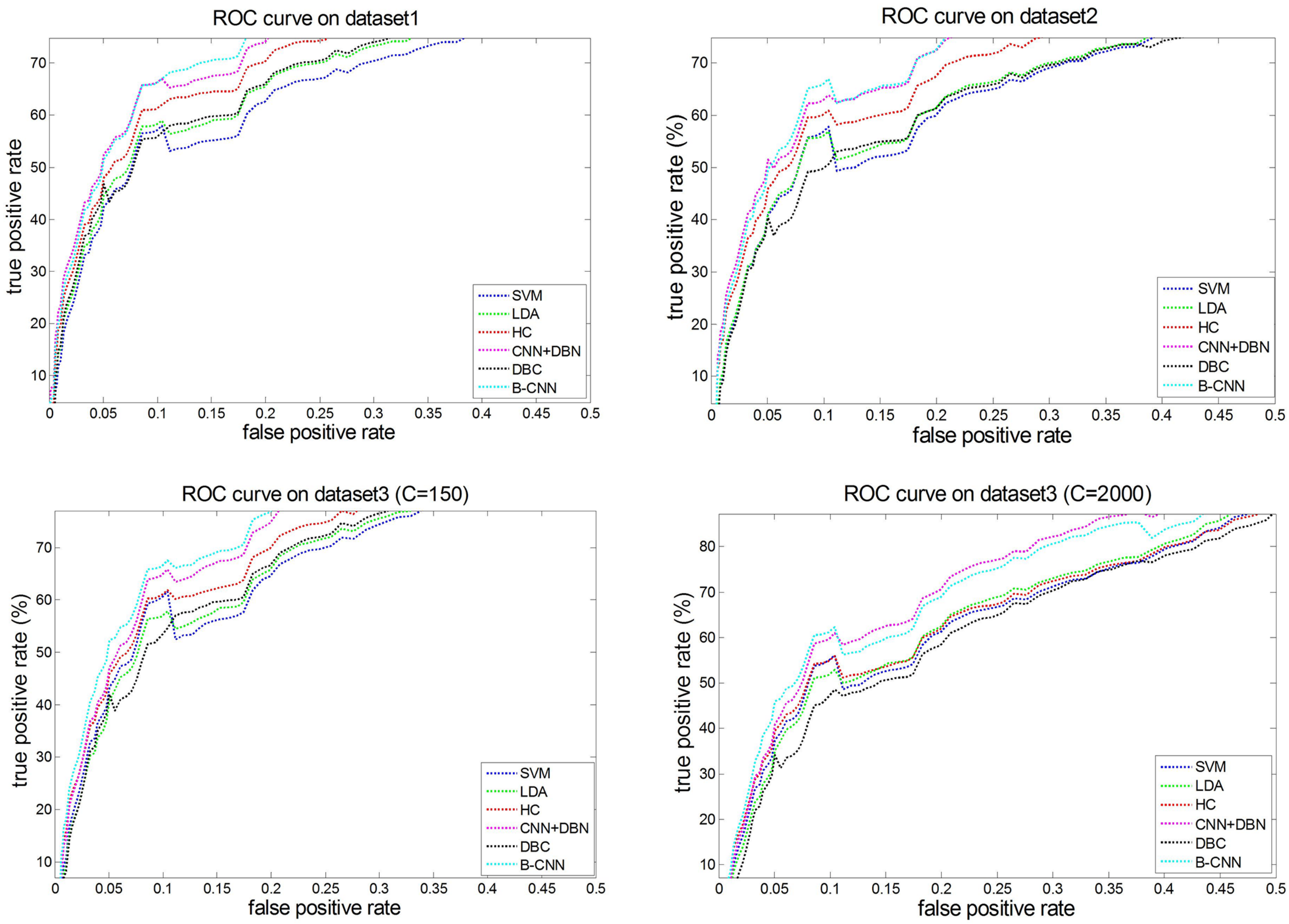

Fig 11. Roc curves on all datasets.

https://doi.org/10.1371/journal.pone.0197933.g011

fluctuation. We analyze the main reason for the decrease. The major reason is also derived from the robustness analysis of these models. When FPR reaches a certain threshold (about *fpr* = 0.1), the TPR performance decreases, and this decrease is significant in dataset 2. We checked the labels of the samples predicted by these models to examine why FPR increased while TPR performance decreased. It appears that the samples with fewer class numbers (sparse category) were not correctly judged. As there were few samples in these categories, the samples could be overlooked when training the classifiers with these models. Thus, when these categories appeared in the test set, the classifiers lost their identification capabilities, leading to downward curve fluctuations. (2) Fig 11 shows that the effects of our model (B-CNN) on dataset 1, dataset 2, and dataset 3 (C = 150) are superior to those produced by other methods (the curve is the closest to the top left corner). However, for dataset 3 (C = 2,000), the B-CNN model outperforms CNN+DBN only when the FPR is less than 0.1. Otherwise, our model performs more poorly than CNN+DBN. We examined these results by checking the predicted label samples and found that under the same feature space, the document representations learned by the CNN+DBN model were more similar within documents (larger cosine). Thus, when the FPR is low, which also means the error rate requirements are small, the B-CNN has

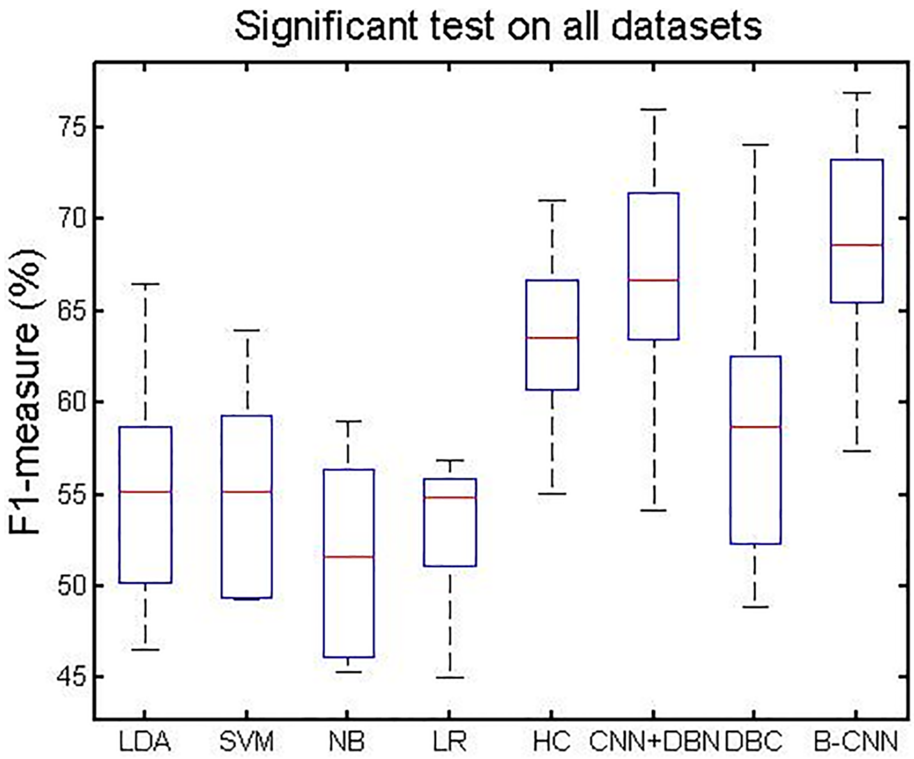

Fig 12. Significant test on all datasets.

https://doi.org/10.1371/journal.pone.0197933.g012

better performance. When FPR increases, the error rate is able to be larger. As the document representation learned by the CNN+DBN model has small differences, making it easier to categorize the documents properly, it is better than the B-CNN model when the FPR is high.

Although our model showed good performance for multi-class and multi-label tasks, and was able to overcome problems caused by unbalanced samples within a certain dimension range, the ROC curve was not always optimal when the number of categories in the unbalanced samples was extremely large (e.g., only 150 of the classes in dataset 3 exceeded 1%). Therefore, we plan to introduce one-shot learning to forecast the characteristics of unbalanced samples in our future research.

5.3.4 Significance test. Fig 12 shows the F_1-measure comparison of performance between our method and other shallow and deep learning approaches. From this figure, we can observe the following. (1) The deep learning methods had better performance than the shallow learning methods. (2) The hierarchical indexing framework was better than the flat learning methods. (3) Our B-CNN method and the CNN+DBN were ranked first and second, respectively, among all of the methods. This indicates that the feature fusion method between global and local entities in each document benefits from the advantages of DBM.

5.3.5 Friedman test. The Friedman test is a non-parametric statistical test and is used to detect differences in treatments across multiple test attempts. The statistic test is calculated by converting the original results to rank results (the best performing algorithm ranks the 1st, the second best one ranks the 2nd, etc.,). Then, the Friedman test needs the computation of the average ranking [79]. Table 6 shows a comparison of performances with the metric F_1-measure among the eight algorithms selected in the experimental study. With the eight algorithms and the four data sets, F_F is distributed according to the F distribution with 8-1 = 7

Table 6. Comparison of F measure (%) among the eight algorithms selected in the experimental study. The ranks are used in the computation of the Friedman test.

Data set	LDA	SVM	NB	LR	HC	CNN+DBN	DBC	B-CNN
dataset 1	46.52 (6)	49.27 (4)	45.34 (7)	45.08 (8)	55.11 (2)	54.13 (3)	48.94 (5)	57.39 (1)
dataset 2	51.43 (6)	49.50 (7)	46.44 (8)	54.86 (4)	62.65 (3)	66.52 (2)	53.45 (5)	68.19 (1)
dataset 3 (C = 150)	56.08 (6)	57.82 (5)	55.50 (8)	55.54 (7)	65.30 (3)	69.97 (2)	58.67 (4)	72.02 (1)
dataset 3 (C = 2000)	66.48 (5)	63.97 (6)	59.01 (7)	56.89 (8)	71.04 (4)	76.03 (2)	74.03 (3)	76.95 (1)
average rank	5.75	5.5	6	6.75	5	3	3.5	1

https://doi.org/10.1371/journal.pone.0197933.t006

and (8-1)*(4-1) = 21 degrees of freedom. According to the $F(7,21)$ distribution, the computed p-value is lower than 0.01, so the null hypothesis is rejected at a high level of significance. From this table, we can observe the following statements. (1) The proposed B-CNN model achieves the best performance (which is ranked the 1st). (2) The DBC and CNN+DBN models performs slightly poorer than the B-CNN model, and the performance of the CNN+DBN model is superior to that of the DBC because the CNN+DBN is based on the hierarchical classification, which can improve the results greatly.

5.4 From biomedicine indexing to newsgroup classification

In Fig 13, we can see that our B-CNN framework achieved good results not only for biomedical indexing but also for other areas of text classification. For dataset 4 (20 newsgroups, C = 20), we choose the LDA extension instead of the entity feature extension, as LDA can be used to represent conventional topics. Using the LDA as the features (this is the global feature in each document) served to enrich the dataset 4 document representation. The experimental results indicated that the B-CNN framework for the 20 newsgroups dataset was slightly better than those of the other methods. We plan to investigate other features relevant to the newsgroups dataset in future research.

Table 7 shows the results for the 20 newsgroups dataset with four classes. We compared the best results obtained from our model with those obtained by Hingmire [80] and Lai [30]. The CNN in this table is described in [29], and we directly quote the results from Lai's paper. We used the word embedding representation on a general dataset rather than the newsgroups dataset. Unsurprisingly, our result was similar to Lai's. Our model performed better than the CNN, which indicates that combining the global features of the two models (merged by the DBM model) can increase the performance of the model. Furthermore, this concept is applicable to other fields.

6 Conclusion

This paper presents a new model (B-CNN) for semantic indexing, which is based on the CNN and can learn semantic representations. We tested this model using the task of indexing biomedical abstracts. Our experimental results demonstrate that the low-dimensional representation of the output layer in our framework is more compact and effective compared with a number of alternative methods. We have found that adaptively grouping word2vec categories into (coarse) subsets via clustering is an effective way to remove noise from unbalanced distributed samples. This is especially the case when dealing with a large number of labels and a massive number of biomedicine documents. Our proposed hierarchical indexing structure achieves effective performance, and can be easily extended to other multi-class indexing tasks. However, our model faces the problem of gradients vanishing when the number of categories in the unbalanced samples is extremely large. The long short term memory network can

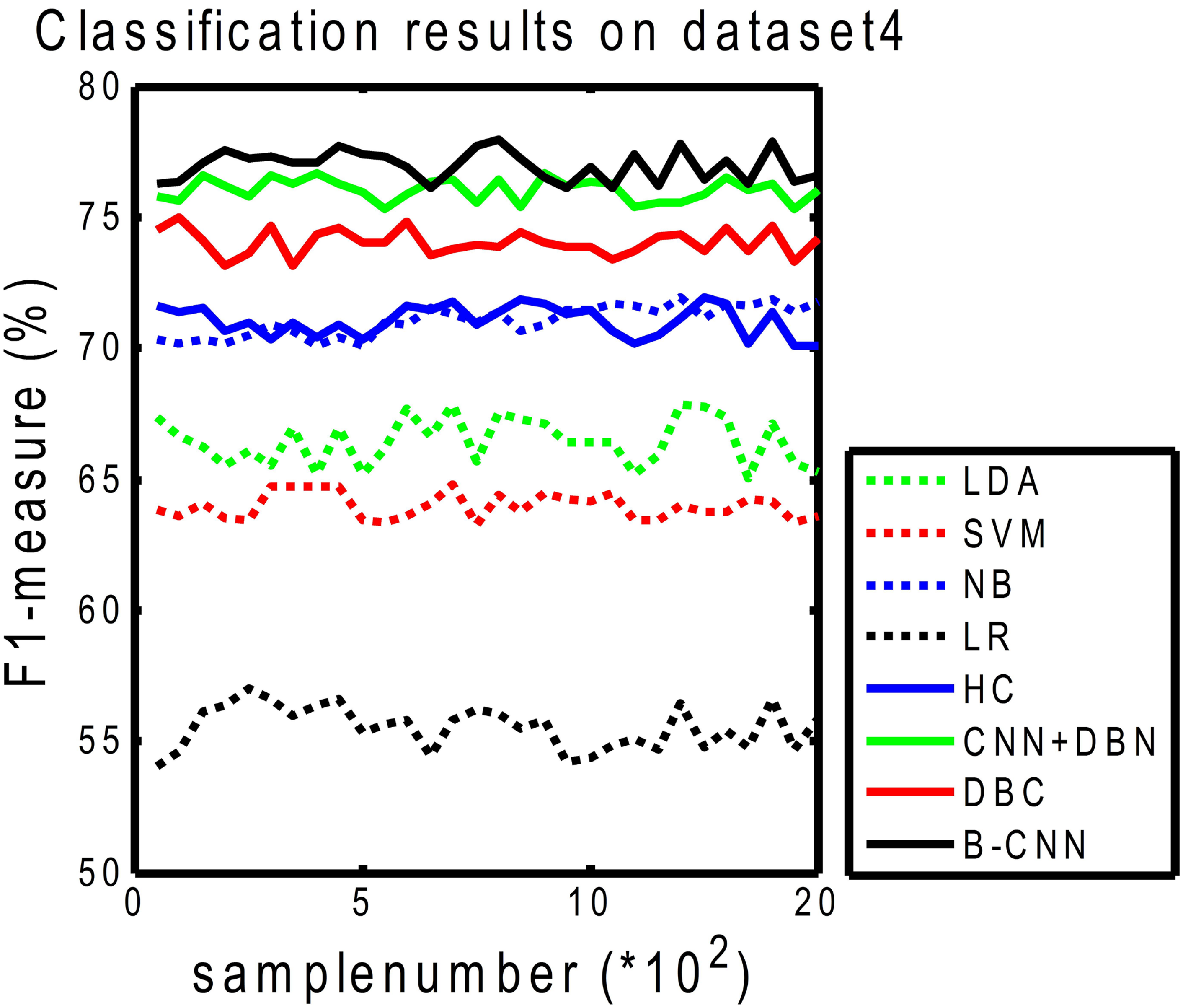

Fig 13. F_1-measure curves on dataset 4 (C = 20).

https://doi.org/10.1371/journal.pone.0197933.g013

effectively learn features and also obtain strong results on a variety of sequence modeling tasks. Therefore, we plan to introduce this model in our future research.

There are three issues in our future research: 1) how to effectively construct the hierarchical indexing label tree with different dependencies among labels to improve the accuracy of multi-

Table 7. Classification results (%) on dataset 4 (C = 4).

Model	20 News
ClassifyLDA-EM [80]	93.60%
RCNN [30]	96.49%
CNN [29]	94.79%
B-CNN	95.13%

https://doi.org/10.1371/journal.pone.0197933.t007 label classification, e.g., label embedding relations (already mentioned in this paper) and co-occurrence relations; 2) how to develop an approach to predict labels with variable lengths in multi-label prediction; and 3) how to extend our approach to apply to image's text classification [81–83].

Author Contributions

Conceptualization: Sujian Li.

Methodology: Yan Yan, Chun Yang, Sujian Li.

Project administration: Yan Yan, Xu-Cheng Yin.

Resources: Yan Yan.

Software: Yan Yan.

Supervision: Xu-Cheng Yin.

Writing – original draft: Yan Yan, Chun Yang, Bo-Wen Zhang.

Writing – review & editing: Yan Yan, Bo-Wen Zhang.

References

1. Salakhutdinov R, Hinton G. Semantic hashing. International Journal of Approximate Reasoning. 2009; 50(7):969–978. https://doi.org/10.1016/j.ijar.2008.11.006
2. Mirowski P, Ranzato M, LeCun Y. Dynamic auto-encoders for semantic indexing. In: NIPS 2010 Workshop on Deep Learning; 2010. p. 1–9.
3. Wu H, Min MR, Bai B. Deep Semantic Embedding. In: SIGIR 2014 Workshop on Semantic Matching in Information; 2014. p. 46–52.
4. Hinton G, Salakhutdinov R. Reducing the dimensionality of data with neural networks. Science. 2006; 313(5786):504–507. https://doi.org/10.1126/science.1127647 PMID: 16873662
5. Xiao T, Xu Y, Yang K, Zhang J, Peng Y, Zhang Z. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. 2014; 40(1):842–850.
6. Ge ZY, Mccool C, Sanderson C, Corke P. Modelling local deep convolutional neural network features to improve fine-grained image classification. In: IEEE International Conference on Image Processing; 2015. p. 4112–4116.
7. Li Y, Lu H, Li J, Li X, Li Y, Serikawa S. Underwater image de-scattering and classification by deep neural network. Computers and Electrical Engineering. 2016; 54:68–77. https://doi.org/10.1016/j.compeleceng.2016.08.008
8. Wang Y, Li J, Gong Y. Small-footprint high-performance deep neural network-based speech recognition using split-VQ. In: IEEE International Conference on Acoustics, Speech and Signal Processing; 2015. p. 4984–4988.
9. Tu Y, Du J, Xu Y, Dai L, Lee CH. Deep neural network based speech separation for robust speech recognition. In: International Conference on Signal Processing; 2015. p. 532–536.
10. Zhou P, Dai L, Liu Q, Jiang H. Combining information from multi-stream features using deep neural network in speech recognition. In: IEEE International Conference on Signal Processing; 2013. p. 557–561.
11. Nie YP, Han Y, Huang JM, Jiao B, Ai-Ping LI. Attention-based encoder-decoder model for answer selection in question answering. Frontiers of Information Technology and Electronic Engineering. 2017; 18(4):535–544. https://doi.org/10.1631/FITEE.1601232
12. Feng M, Xiang B, Glass MR, Wang L, Zhou B. Applying deep learning to answer selection: A study and an open task. In: IEEE Workshop on ASRU. IEEE; 2015. p. 813–820.
13. Severyn A, Moschitti A. Learning to rank short text pairs with convolutional deep neural networks. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM; 2015. p. 373–382.
14. Dahl GE, Yu D, Deng L, Acero A. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. Audio, Speech, and Language Processing, IEEE Transactions on. 2012; 20(1):30–42. https://doi.org/10.1109/TASL.2011.2134090

15. Collobert R, Weston J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th international conference on Machine learning. ACM; 2008. p. 160–167.

16. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems; 2012. p. 1097–1105.

17. Jimeno-Yepes A, Mork JG, Demner-Fushman D, Aronson AR. Comparison and combination of several MeSH indexing approaches. In: AMIA 2013, American Medical Informatics Association Annual Symposium; 2013.

18. Karami A, Gangopadhyay A. FFTM: A Fuzzy Feature Transformation Method for Medical Documents. In: ACL 2014 Workshop on BioNLP; 2014. p. 128–133.

19. Aronson A, Bodenreider O, et al. From indexing the biomedical literature to coding clinical text: experience with MTI and machine learning approaches. In: ACL 2007 Workshop on BioNLP: Biological, Translational, and Clinical Language Processing; 2007. p. 105–112.

20. Fieschi M, et al.; IOS Press. Application of a medical text indexer to an online dermatology atlas. Medinfo. 2004; 107:287.

21. Trieschnigg D, Pezik P. MeSH Up: effective MeSH text classification for improved document retrieval. Bioinformatics. 2009; 25(11):1412–1418. https://doi.org/10.1093/bioinformatics/btp249 PMID: 19376821

22. Jessen WJ, Landschulz KT, Turi TG, Reams RY. Mining PubMed for Biomarker-Disease Associations to Guide Discovery. Nature Precedings. 2015;.

23. Drakopoulos G, Kanavos A. Tensor-based document retrieval over Neo4j with an application to PubMed mining. In: International Conference on Information, Intelligence, Systems and Applications; 2016.

24. Reynolds C, Matteis SD, Cullinan P, Szram J. Pubmed mining for occupational idiopathic pulmonary fibrosis papers. In: Ers International Congress; 2017. p. PA406.

25. Zhou J, Fu BQ. The research on gene-disease association based on text-mining of PubMed. Bmc Bioinformatics. 2018; 19(1):37. https://doi.org/10.1186/s12859-018-2048-y PMID: 29415654

26. dos Santos CN, Gatti M. Deep convolutional neural networks for sentiment analysis of short texts. In: COLING; 2014. p. 69–78.

27. Yepes AJ, MacKinlay A, Bedo J, Garnavi R, Chen Q. Deep Belief Networks and Biomedical Text Categorisation. In: Proceedings of the Twelfth Annual Workshop of the Australasia Language Technology Association. vol. 123; 2014.

28. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998; 86(11):2278–2324. https://doi.org/10.1109/5.726791

29. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa PP. Natural Language Processing (Almost) from Scratch. Journal of Machine Learning Research. 2011; 12:2493–2537.

30. Lai S, Xu L, Liu K, Zhao J. Recurrent convolutional neural networks for text classification. In: the 29rd AAAI Conference on Artificial Intelligence; 2015. p. 2267–2273.

31. Salakhutdinov R, Hinton G. Deep boltzmann machines. In: International Conference on Artificial Intelligence and Statistics; 2009. p. 448–455.

32. Deerwester S, Dumais S, et al. Indexing by latent semantic analysis. JASIS. 1990; 41(6):391–407. https://doi.org/10.1002/(SICI)1097-4571(199009)41:6%3C391::AID-ASI1%3E3.0.CO;2-9

33. Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. Journal of Machine Learning Research. 2003; 3:993–1022.

34. Hofmann T. Probabilistic latent semantic indexing. In: SIGIR; 1999. p. 50–57.

35. Steyvers M, Griffiths T. Probabilistic topic models. Handbook of latent semantic analysis. 2007; 427 (7):424–440.

36. Mcauliffe JD, Blei DM. Supervised topic models. In: NIPS; 2008. p. 121–128.

37. Bai B, Weston J, Grangier D, Collobert R, Sadamasa K, Qi Y, et al. Supervised semantic indexing. In: CIKM; 2009. p. 187–196.

38. Zhang S, Yao H, Sun X, Wang K, Zhang J, Lu X, et al. Action recognition based on overcomplete independent components analysis. Information Sciences. 2014; 281:635–647. https://doi.org/10.1016/j.ins.2013.12.052

39. Lee H, Grosse R, Ranganath R, Ng AY. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the 26th Annual International Conference on Machine Learning. ACM; 2009. p. 609–616.

40. Zeng D, Liu K, Lai S, Zhou G, Zhao J. Relation classification via convolutional deep neural network. In: Proceedings of COLING; 2014. p. 2335–2344.

41. Socher R, Karpathy A, Le QV, Manning CD, Ng AY. Grounded compositional semantics for finding and describing images with sentences. Transactions of the Association for Computational Linguistics. 2014; 2:207–218.

42. Read J, Perez-Cruz F. Deep Learning for Multi-label Classification. Machine Learning. 2014; 85 (3):333–359.

43. Yang Y, Yang Y, Yang Y, Yang Y. Deep Learning for Extreme Multi-label Text Classification. In: International ACM SIGIR Conference on Research and Development in Information Retrieval; 2017. p. 115–124.

44. Mori K, Inoue N, Shinoda K. D-12-10 FEATURE DIMENSION REDUCTION FOR VIDEO SEMANTIC INDEXING USING DEEP LEARNING. In: Proceedings of the IEICE General Conference; 2014.

45. Inoue N, Shinoda K. Vocabulary Expansion Using Word Vectors for Video Semantic Indexing. In: ACM International Conference on Multimedia; 2015. p. 851–854.

46. Podlesnaya A, Podlesnyy S. Deep Learning Based Semantic Video Indexing and Retrieval. In: Sai Intelligent Systems Conference; 2016. p. 359–372.

47. Nie L, Akbari M, Li T, Chua TS. A Joint Local-Global Approach for Medical Terminology Assignment. In: MedIR@ SIGIR; 2014. p. 24–27.

48. Nie L, Li T, Akbari M, Shen J, Chua TS. Wenzher: Comprehensive vertical search for healthcare domain. In: Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval. ACM; 2014. p. 1245–1246.

49. Nie L, Wang M, Zhang L, Yan S, Zhang B, Chua TS. Disease inference from health-related questions via sparse deep learning. IEEE Transactions on Knowledge and Data Engineering. 2015; 27(8):2107–2119. https://doi.org/10.1109/TKDE.2015.2399298

50. Shen Y, He X, Gao J, Deng L, Mesnil G. Learning semantic representations using convolutional neural networks for web search. In: the 23rd international conference on World wide web companion. International World Wide Web Conferences Steering Committee; 2014. p. 373–374.

51. Johnson R, Zhang T. Effective use of word order for text categorization with convolutional neural networks. arXiv preprint arXiv:14121058, (2014). 2014;.

52. Salakhutdinov R, Larochelle H. Efficient learning of deep Boltzmann machines. In: International Conference on Artificial Intelligence and Statistics; 2010. p. 693–700.

53. Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. Neural computation. 2006; 18(7):1527–1554. https://doi.org/10.1162/neco.2006.18.7.1527 PMID: 16764513

54. Mikolov T, Zweig G. Context dependent recurrent neural network language model. In: Spoken Language Technology Workshop; 2012. p. 234–239.

55. Settles B. Biomedical named entity recognition using conditional random fields and rich feature sets. In: NLP 2004 Workshop on Biomedicine and its Applications; 2004. p. 104–107.

56. Glorot X, Bordes A, Bengio Y. Deep sparse rectifier networks. In: the 14th International Conference on Artificial Intelligence and Statistics. JMLR. vol. 15; 2011. p. 315–323.

57. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:12070580. 2012;.

58. Thabtah F, Cowling P, Peng Y, et al. MMAC: A new multi-class, multi-label associative classification approach. In: Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on. IEEE; 2004. p. 217–224.

59. Turian J, Ratinov L, Bengio Y. Word representations: a simple and general method for semi-supervised learning. In: ACL 2010; 2010. p. 384–394.

60. Hu X, Zhang X, Lu C, Park EK, Zhou X. Exploiting Wikipedia as external knowledge for document clustering. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM; 2009. p. 389–396.

61. Ananiadou S, Friedman C, Tsujii J. Introduction: named entity recognition in biomedicine. Journal of Biomedical Informatics. 2004; 37(6):393–395. https://doi.org/10.1016/j.jbi.2004.08.011

62. McCallum AK, et al. A machine learning for language toolkit. MALLET. 2002; 15:131–136.

63. Mikolov T, Chen K, Corrado G, Dean J. Efficient Estimation of Word Representations in Vector Space. CoRR, 2013. 2013;abs/1301.3781.

64. Mikolov T, Yih Wt, Zweig G. Linguistic Regularities in Continuous Space Word Representations. In: HLT-NAACL; 2013. p. 746–751.

65. Hu X, Zhang X, Lu C, Park EK, Zhou X. Exploiting Wikipedia as external knowledge for document clustering. In: KDD 2009; 2009. p. 389–396.

66. Sokolova M, Lapalme G. A systematic analysis of performance measures for classification tasks. Information Processing & Management. 2009; 45(4):427–437. https://doi.org/10.1016/j.ipm.2009.03.002

67. Jung C, Kim C. A unified spectral-domain approach for saliency detection and its application to automatic object segmentation. Image Processing, IEEE Transactions on. 2012; 21(3):1272–1283. https://doi.org/10.1109/TIP.2011.2164420

68. Huang SC, Chen BH. Highly accurate moving object detection in variable bit rate video-based traffic monitoring systems. Neural Networks and Learning Systems, IEEE Transactions on. 2013; 24 (12):1920–1931. https://doi.org/10.1109/TNNLS.2013.2270314

69. Huang SC. An advanced motion detection algorithm with video quality analysis for video surveillance systems. Circuits and Systems for Video Technology, IEEE Transactions on. 2011; 21(1):1–14. https://doi.org/10.1109/TCSVT.2010.2087812

70. Fawcett T. ROC graphs: Notes and practical considerations for researchers. Machine learning. 2004; 31:1–38.

71. Mimno DM, McCallum A. Topic Models Conditioned on Arbitrary Features with Dirichlet-multinomial Regression. CoRR, 2012. 2012;.

72. Chang C, Lin C. LIBSVM: A library for support vector machines. ACM TIST. 2011; 2(3):27.

73. Chang CC, Lin CJ. LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST). 2011; 2(3):27.

74. Ng AY, Jordan MI. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. In: NIPS 2001; 2001. p. 841–848.

75. Montavon G, Müller KR. Deep Boltzmann machines and the centering trick. In: Neural Networks: Tricks of the Trade. Springer; 2012. p. 621–637.

76. Reed S, Lee H, Anguelov D, Szegedy C, Erhan D, Rabinovich A. Training deep neural networks on noisy labels with bootstrapping. arXiv preprint arXiv:14126596, (2014). 2014;.

77. Mishra AK, Aloimonos Y, Cheong LF, Kassim A, et al. Active visual segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions On. 2012; 34(4):639–653. https://doi.org/10.1109/TPAMI.2011.171

78. Elkan C, Noto K. Learning classifiers from only positive and unlabeled data. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM; 2008. p. 213–220.

79. García S, Fernández A, Luengo J, Herrera F. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences. 2010; 180(10):2044–2064. https://doi.org/10.1016/j.ins.2009.12.010

80. Hingmire S, Chougule S, Palshikar GK, Chakraborti S. Document classification by topic labeling. In: SIGIR 2013; 2013. p. 877–880.

81. Yin XC, Zuo ZY, Tian S, Liu CL. Text Detection, Tracking and Recognition in Video: A Comprehensive Survey. IEEE Transactions on Image Processing. 2016; 25(6):2752–2773. https://doi.org/10.1109/TIP.2016.2554321

82. Yin XC, Pei WY, Zhang J, Hao HW. Multi-orientation scene text detection with adaptive clustering. IEEE transactions on pattern analysis and machine intelligence. 2015; 37(9):1930–1937.

83. Yin XC, Yin X, Huang K, Hao HW. Robust text detection in natural scene images. IEEE transactions on pattern analysis and machine intelligence. 2014; 36(5):970–983. https://doi.org/10.1109/TPAMI.2013.182 PMID: 26353230