Hierarchical Classification with Convolutional Neural Networks for Biomedical Literature

Yan Yan

Department of Computer Science and Technology, University of Science and Technology Beijing

yanyanustb@126.com

ABSTRACT

Multi-label document classification is a challenge task in many real-world applications. Recently, hierarchical classification methods have been widely used in document classification. However, at each layer of the hierarchical architecture, a classifier is trained independently, ignoring the relations between the other layers. In addition, compared with general documents, the biomedical literature only consists of the title and abstract information instead of the whole context. To overcome this problem, in this paper, we propose a novel hierarchical indexing method with Convolutional Neural Networks (CNNs) to tackle with the biomedical abstract document collections. First, we construct a hierarchical CNN indexing architecture which adaptively groups word2vec categories into (coarse) subsets by clustering. Next, a suitable loss function is designed for CNN training, where multi-label classification is actually performed in a coarse-to-fine learning style. Thereafter, a high-dimensional space representation is generated with feature extension by word sequence embedding, which contains more semantic information than bag-of-words. Experimental results show that our CNN model achieves an impressed performance.

Keywords: *Hierarchical Classification, Deep Learning, Convolutional Neural Networks, Biomedical Literature, Semantic Indexing.*

1. INTRODUCTION

With the exponential growth of information on the internet, it is becoming increasingly difficult to find and organize relevant materials. Moreover, many real world classification systems have complex hierarchical structures (e.g., MeSH, Open Directory Project and Yahoo! Directories) [1]. It is a great challenge to classify the document effectively.

There are two mainstream approaches for document classification. One is the flat classification method, which assumes that all labels are independent from each other [2]. Although this method is simple and commonly used, it suffers from the unbalanced data problem. In order to better address this unbalance phenomenon, we transform it into a local equilibrium data by using the hierarchical classification method. Traditional hierarchical classification methods [1,3,4,5,6,7] consider the dependencies of labels, and the order of document labels are rearranged in accordance with the hierarchical tree. This method can not only solve the problem of unbalanced data, but also greatly reduce the dimensions of forecasts. However, each classifier is trained separately, ignoring the relationship between the connected nodes.

These hierarchical classification methods are mostly based on the tree structure, which can be seen as a pruned network [8]. Different from the general network, the parent nodes in the label tree are only connected with their child nodes, i.e. the weights between unconnected nodes are regarded as 0 in the label network. Nowadays, some state-of-the-art artificial neural network methods, such as Deep Belief Networks (DBNs) [9] and Convolutional Neural Networks (CNNs) [10] are widely used for classification problems [11,12]. In this paper, we use CNNs to train our hierarchical model.

In this paper, we propose a new hierarchical classification method based on CNNs for multi-label document literature. First, we transform the unbalanced data to partial equilibrium problem based on the hierarchical classification learning. We combine label embedding with label correlation to construct the label tree. Second, we train the model as a whole to update the weight. In addition, the feature representation is not good enough, since it is only exacted from the abstract information. To solve the problem, we enrich the document representation by using word sequence embedding, Wikipedia categories and entity classes. Empirical results also verify that this improved representation is more compact than Bag-of-Words (BOW).

2. RELATED WORK

2.1 Hierarchical Classification

Generally speaking, hierarchical classification has two steps. One is tree construction. To accommodate the semantics imposed by the hierarchical structure, some researchers have imposed statistical similarity constraints between the probabilistic models for adjacent vertices in the hierarchy. In order to obtain robust parameter estimates, Xue [1] proposed a deep classification approach to categorize Web documents which contains two stages: search stage and classification stage. In the first stage, for a given document, use a category-search algorithm acquire the category candidates, in order to creating a dynamic smaller hierarchy. Then in the second stage focus on this small subset of the original hierarchy. Sun [7] proposed a method that Category-Similarity Measures and Distance-Based Measures to consider the degree of misclassification in measuring the classification performance. The general judgment of the similarity of the labels was calculated by the cosine distance, but this kind of character that represents the label with the participation of a large artificial selection. As a consequence, it is very difficult to conduct this process on the large number of categories.

The other is how to train the classifiers based on the hierarchical tree. [3,6] explored hierarchical classification method for web page. These hierarchical structures were initially used to train different second-level classifiers by SVM method. Then the web label was predicted through combined scores from the top and second-level models using different combination rules. Gopal [4,5] proposed an algorithmic framework for large scale hierarchical classification learning based on predefined regularization structure. This framework associated a parallel iterative optimization scheme with objective in the hierarchy and also combines ideas from large margin kernel methods and Bayesian analysis to train classifiers. Experimental results demonstrated that the approach was effective on synthetic, text and speech data. Lin [13] proposed a hierarchical framework based on ensemble classification methods. In the first layer train a set of base classifiers by various selection strategies, divides the input into seven classes; in the second layer of which, an analogous ensemble method which combines the selected base classifiers based on clustering is adopted to predict all protein folds.

Traditional hierarchical classification algorithms construct classifiers at each level of the category tree where each classifier works as a flat classifier at that level. They only focused on building classifiers trained separately, ignoring the relationship between the connected nodes contacts.

2.2 Deep Neural Network

In the last several years, deep neural networks (DNNs) [9] have emerged as a powerful machine learning technology, which achieves tremendous success for image classification, speech recognition and natural language processing (NLP) tasks, by showing significant gains over the state-of-the-art shallow learning. Deep learning techniques, e.g., DBNs and CNNs, are about to be explosively applied in the research fields of information retrieval and natural language processing.

Hinton and Salakhutdinov [9] introduced a moderately fast, unsupervised learning algorithm for deep models called Deep Belief Networks. The DBN can be viewed as a composition of stacked Restricted Boltzmann Machines (RBMs) that contain visible units and hidden units. The visible units represent the document data and the hidden units represent features learned from the visible units. Then they proposed a two-layer undirected graphical model [14] called “Replicated Softmax model” (RSM) composed with RBMs to explore the use of basic deep learning methods to represent the document information, which had a better result than LDA method. Recently, some researchers applied CNNs to several NLP tasks and achieved considerable effects. For example, Zeng et. al. used CNNs for relation classification [12], and Dos Santos utilized CNNs for semantic analysis of text [11]. For document, Convolutional Neural Network consists of convolution layers, ReLU and k-max pooling layers. In convolution networks, each neuron is only connected to the local area of the lower layer instead of the whole layer of neurons. That is, it will extract partial information or features [10]. ReLU is the abbreviation of Rectified Linear Unit, which is used to increase the nonlinear properties of network as well as sparsity without affecting the receptive fields of the convolution layer. Max pooling acts as feature mapping layer and its operator is a non-linear subsampling function that returns the maximum of a set of values. At the end of the network there are the fully-connected layers where each feature map connects to all sub-sampling maps from the previous layer.

2.3 Conventional Biomedical Classification

In biomedical document classification task, a huge number of semantic labels from automatically annotating MeSH terms for Medline citations which will face the extreme unbalanced data. In order to address this problem, Phoungphol [15] formulated a multi-class Support Vector Machine (SVM) model to improve the biomedical document classification performance. This model used cost-sensitive approach and ramp loss function to address the imbalanced data problem and can work effectively on 7 real-world imbalanced biomedical datasets. Yepes [16] used (DBN) to automatically generate a new representation in biomedical text categorization. Through the hidden neurons extracting relevant features from the observations, as a consequence, the DBN learn to extract a deep hierarchical representation of the training data. The experimental results showed that the DBN on biomedical document classification superior performance compared to SVM method. However, for the biomedical abstract document, a massive collection with only the title and abstract information which ignore in these methods. But there is still room for improvement by combine the biomedical domain knowledges.

3. METHODOLOGY

Considering the numerous classes of the documents and the unbalanced distributions of samples, we introduce a hierarchical CNN-based framework in order to solve multi-class and multi-label semantic indexing with correlated labels for biomedicine document. The architecture of our proposed framework is showed in Figure1. We mainly modify two parts (input and output) of the conventional CNN model to adapt this task. We design a novel vector representation based on words sequence information for biomedicine documents to improve the feature representation (input). This helps us to take advantage of CNN's own strength which is more suitable to deal with the features with sequence relationship. The other is a hierarchical CNN-based classification framework (CNN's output), which is much better than flat classification in processing large number of classes. We generate a label tree based on label embedding and label correlation. Moreover, the coarse clustering step is an effective way to remove noise from the uneven distributed samples. In addition, we also design suitable loss functions for the learning process of this framework.

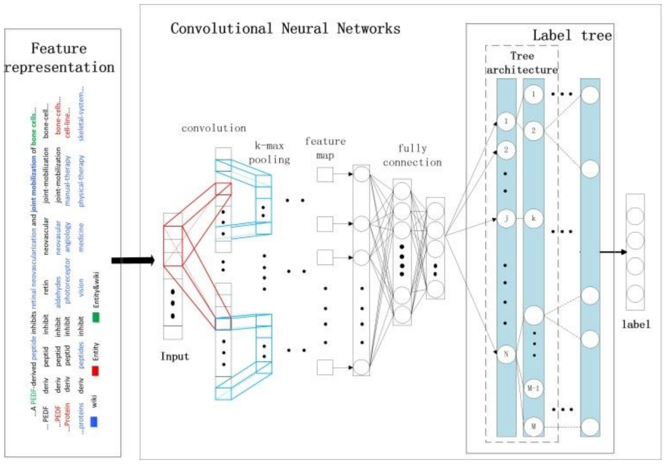
Feature representation
...A PEDF-derived peptide inhibits retinal neovascularization and joint mobilization of bone cells...
... PEDF deriv peptid inhibit retin neovascular joint-mobilization bone-cell...
...PEDF deriv peptid inhibit aldehydes neovasular joint-mobilization bone-cells...
...Protein deriv peptid inhibit photoreceptor angiology manual-therapy cell-line...
...proteins deriv peptides inhibit vision medicine physical-therapy skeletal-system...
🟦 wiki 🟥 Entity 🟩 Entity&wiki
Convolutional Neural Networks
convolution
k-max pooling
feature map
fully connection
Input
Label tree
Tree architecture
1
2
j
N
1
2
k
M-1
M
label

Fig. 1. A hierarchical CNN-based framework for semantic indexing.

3.1 Model Implement with Label Tree

In general, labels are not independent. Especially the more categories, the richer information. So effectively using the dependencies among labels can improve the accuracy of multi-label classification. Besides, the interrelationship between the labels themselves, each document can have more than one topic. Between the topics there are also some potential echoing relationships. One is the relationship between some pairs of tags that are included each other. If we know the tags' parent node, we will improve the probability of the child nodes of the parent node corresponding to the category to some extent because the dimensions of categories are greatly reduced. At the same time, it can overcome the weakness of the classifiers distinguishing these categories caused by small category of fewer samples to a certain extent, and can also improve the overall classification accuracy. The other one is that some labels have parallel relations, and the topics do not appear in the same document. It can also greatly save the time for the classifier to loop through all the labels.

In this paper, we design an indexing architecture (shown in hierarchical classification part of Figure 1) that introduces label (word) embedding trained by biomedicine corpus for classification. That is, we have two tree-struct layers in the label tree. The first is top-level category which is coarse classification and also the parent nodes we have defined. The second is sub-level category which is leaf nodes. We use two parts of information to construct the hierarchy category tree. The first is synonymous relations of labels, and the second is the labels co-occurrence probabilities.

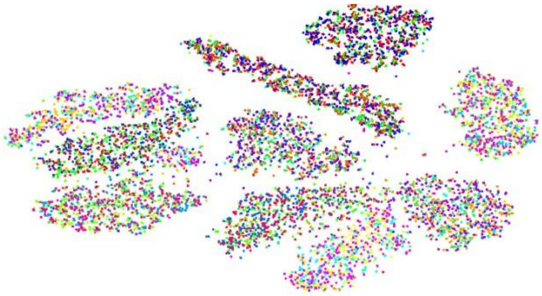

Fig. 2 Coarse clusters of label embedding.

The word vector model can be seen as a language model that consists of n-dimensional continuous valued vectors, and each dimension of the word embedding expresses a latent feature of the words, expectably reflecting useful semantic and syntactic regularities [17]. In this paper, we use word embedding to mine synonyms relations of labels. Figure2 shows the label embedding representation of coarse clusters (partly). Through the label synonymous relationship we can get a good clustering about the labels. But because of the particularity of the biomedicine text, we check the labels and find that many of these labels also have subordination relationship. So we use labels co-occurrence probability to mine the dependent and subordinate relationship between the tags in the training samples, and thus to learn the tree structure of relationships hiding between labels.

The details are as follows. The label is represented with the word embedding and calculated the distance between labels. Words with similar meaning or words that have dependency relationship will have a close range in one dimension space. We count the probability of every label appearing under another labels from the training samples, then combine the probability value of the two parts, finally get the hierarchy tree. We visualize the labels clustering results in two-dimensional space and obtain good results. At this point, all the labels are divided into two levels, then we put labels on all the samples in accordance with the order of the tree structure.

3.2 Model Expanding by Label Tree

The traditional classification approach is flat classification method, we introduce the tree-architecture to implement the overall hierarchical learning. The tree-struct is designed in order to make full use of the label tree, and improve the classification performance.

In this paper, our hierarchical CNN-based framework consist of convolution layers, pooling layers, fully-connected layers and tree-architecture layers. The tree-architecture is shown in Figure3. The layer's input is the feature representation and in our model is fully-connected layer feature of CNN's. We can learn this feature by other neural networks. The layer's output is the predict labels of the document. The nodes with the same depth of label tree are organized in one layer. Two connected layer is connected by the tree-architecture.

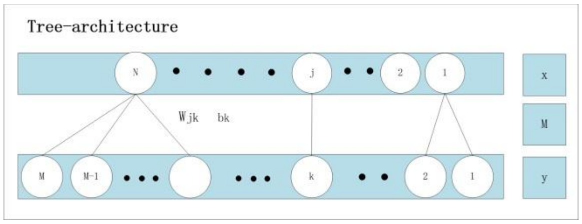

Fig. 3 Tree-architecture.

In the tree-architecture, N represents the parent nodes number and M represents all child nodes number. x is the output of fully-connected layers, and $x \in R^{N \times 1}$. y is the label of document (label layer) and $y \in R^{M \times 1}$. M is the mask matrix which is used to reflect the architecture of the hierarchical tree and $M \in \{0,1\}^{M \times N}$. M_{jk} is regard as 1 when node x_j and y_k is connected, otherwise is 0. W is the weights of the connected two layers and $W \in R^{M \times N}$. b is the biases and $b \in R^{M \times 1}$.

Formally, the label is calculated by $y = \phi(W, M)x + b$, where $\phi(W, M)$ represent the element-wise multiplication of matrix W and M. (i.e. if $S = \phi(W, M), S_{ij} = W_{ij} M_{ij}$)

We use $\frac{\partial z}{\partial y}$ (where $\frac{\partial z}{\partial y} \in R^{M \times 1}$) to represent the derivation of the CNN output w.r.t. y, and

$$\frac{\partial z}{\partial W} = \phi(x(\frac{\partial z}{\partial y})^T, M) \tag{1}$$

$$\frac{\partial z}{\partial b} = \frac{\partial z}{\partial y} \tag{2}$$

$$\frac{\partial z}{\partial x} = \phi(W^T \frac{\partial z}{\partial y}, M^T) \tag{3}$$

By computing derivatives of the model using back-propagation, the weights and biases can be updates as

$$W(t) = W(t-1) - \eta\phi(\frac{\partial z}{\partial W}, M) \tag{4}$$

$$b(t) = b(t-1) - \gamma \frac{\partial z}{\partial b} \tag{5}$$

We use softmax loss function and back-propagation to update the weights in the model. The relationship of connected nodes is represented by the mask matrix M in the tree-architecture layer. The number of tree-architecture layers is determined by the structure of the label tree. We select two-layer tree-architecture in our paper. The first layer is also the coarse classification, which is a multi-class classification problem. These labels are independent in this layer, which belongs to only one category per test sample.

3.3 Document representation with word Sequence Embedding (DSE)

In this paper, we design a novel vector representation based on word sequence information for biomedicine documents. The sequence of words in the documents as the basic feature, considering the indexing task of biomedical literature, we use Wikipedia categorial information and entity meta-features based on Metamap keywords in biomedicine fields as synonym for expansion. MetaMap [18] is a widely applied open source toolkit which extracts concepts in the UMLS[1] metathesaurus.

We preprocess all documents through removing stop-words and stemming, then the rest words arranged in the order of the original documents. In order to fix the length of all documents, we make the following process as algorithm1.

```
Algorithm 1: Document word Sequence Embedding (DSE)
Input: $D$
Output: document representation $V$
L=average length of $D$
for $k=1…N$ do
    $m$=length($D_k$)
    if $m<L$
        $V=[V_{k1},V_{k2},...,V_{km},Unknown_1,...,Unknown_{L-M}]$;
    Else if $m=L$
        $V=[V_{k1},V_{k2},V_{k3},...,V_{km}]$;
    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 $L$. $V=[V_{newk1},V_{newk2},...,V_{newkm},...,V_{newkL}]$;
    end if
end for
Return $V$
```

D represent the documents after the process of removing stop-words and stemming; D_k represent document k; Unknown indicated the word "Unknown" which has an effective word embedding in the large corpus. When the document length is less than L, we use this word to supplement the length. "L-m" in $Unknown_{L-m}$ means the number of Unknown words, that is, when the document has only one Unknown word, we just use $Unknown_1$; V_{km} means the word embedding of the m word in document k and V_{newkm} means the m word embedding in new words sequence after rearranged in the document.

Entity and Wikipedia as meta-feature enrich document representation: in this paper, we recognition entity class in biomedicine abstract use a Machine Learning for Language Toolkit [2] implementation(MALLET) of CRFs and the entity class also used in the hierarchical indexing framework.

In addition, we use the Wikipedia [19] categories of the corresponding anchor words based on Metamap keywords in biomedicine literature to enrich document representation. For each word in the V we select three words to enrich document representation from the corresponding categories and entities. These extension of words orderly follow the original words. Then word embedding is used to represented each word. Word embedding [17,20] can be seen as a language model that consists of n-dimensional continuous valued vectors, and each dimension of the word embeddings expresses a latent feature of the words, expectably reflecting useful semantic and syntactic regularities. When the number of the word categories is less than three, we use the word itself in the document together with its categories.

We introduce a concrete example to describe the extend process in details. For example, there is one sentence in the document (shown in figure4). "... A PEDF-derived peptide inhibits retinal neovascularization and joint mobilization of bone cells...". "A", "and", "of" are stop-words, "peptide", "retinal", "neovascularization" and "joint mobilization" are anchor words, they all have corresponding wikipedia categories and are marked with blue colour. "PEDF" and "bone cell" are name entity as well as anchor words with Wikipedia category. They belong to the Protein and cell-line properties respectively, marked with green colour. The word "peptide" has one corresponding wikipedia category, "retinal" and "neovascularization" have three corresponding, and "joint mobilization" has two.

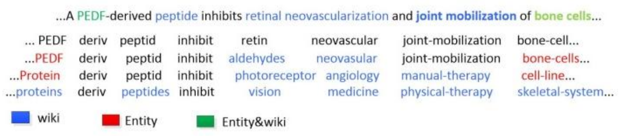
...A PEDF-derived peptide inhibits retinal neovascularization and joint mobilization of bone cells...
... PEDF deriv peptid inhibit retin neovascular joint-mobilization bone-cell...
...PEDF deriv peptid inhibit aldehydes neovasular joint-mobilization bone-cells...
...Protein deriv peptid inhibit photoreceptor angiology manual-therapy cell-line...
...proteins deriv peptides inhibit vision medicine physical-therapy skeletal-system...
wiki
Entity
Entity&wiki

Fig. 4. sentence representation.

The expanded rules are as follows. If the word is both entity and anchor, for example, "PEDF" and "bone cells", the extending order is that retain the original information of the word first and then use this word's entity extension (marked use red colour), at last extend

[1] http://www.nlm.nih.gov/research/umls/implementation_resources/metamap.html

[2] http://mallet.cs.umass.edu/

with wikipedia categories of information. If it's just anchor word and the word has less than three corresponding categories, the first expansion uses the original word, and then the category expansion, e.g. the word “joint-mobilization” After preprocessing, the sentence is represented by the word embedding that every word is represented with a 50-dimensional vector, which also means that each document is represented as a 200*L matrix.

4. EXPERIMENTS

4.1 Dataset and Experimental Setup

The dataset1[1] is a labeled corpus of English scientific medical abstracts from Springer website. There are 39 classes with 250 samples in each class on average. The dataset2[2] is a BioText data and there are 168 classes in 1000 documents. The dataset3[3] is MEDLINE dataset. Yepes’ paper [16] selected the top 10 most frequent MeSH headings to avoid the extremely unbalanced distribution dataset. As same as their motivation, we select the top 150 MeSH headings because only about 150 appear in more than 1% of the whole medline. Also in order to extend the experiments, in our paper, we introduce some unbalanced samples and select the top 2000 categories. We processed the citations to extract the text from the title and the abstract. The former two datasets are multi-class and the third is multi-label. We randomly select 80% as the training samples and 20% as the test samples, and make 10 times of cross validation in experiment on all datasets. We apply the widely-used cuda-convnet package to train our model on a single GPU.

4.2 Details

We conducted three parts of experiments. In the first part we compare DSE with other feature representation. In the second part we analyze different models' effect with these features representation and significant test is described on these datasets. In the third part, we compared our model with DBN model on all datasets.

Based on our DSE feature extension, we also enrich BOW representation through Wikipedia and entity class based on the Metamap, and denote this as BOW+ as our baseline. In addition, we make some changes when training CNN. Since our purpose is to learn the structure and semantic information between words, so we fix the size of the slide window to make it equal to dimension of word embedding (50-dimension). We step down sliding window position 50 by 50, so that each sliding window does not change the word vector embedding which represents the word's feature, learning the information of the document by changing the length of sliding window.

Pattern Matching (**PM**) [21]: by comparing MeSH labels with words in the documents to predict document label. During the matching process, we add some artificial rules: convert all words of MeSH label into lowercase and make some fuzzy matching. **MTI [22]** is a document indexing system, providing recommendations based on the MeSH and MEDLINE database. The **MeshUP (MU)** system is developed by [23] exploring the combination of different machine learning approaches to perform classification on the full class-set. **CNN_flat**: for each class, a binary classification is utilized to extend each node into two nodes. **DBN_flat** and **DBN_H** represent the flat classification and hierarchical classification based on DBN model. Besides, we also select Latent Dirichlet Allocation (**LDA**) [24], Naive Bayesian (**NB**) [25], Logistic Regression (**LR**) [26] and Supported Vector Machines (**SVM**)[4] [27] compare with our hierarchical classification (**CNN_H**) method.

We compare the performance of our best model (with five pooling layers, each of which follows a convolution layer, three fully-connected layers and two tree-struct layers at the end of CNNs) with typical methods. We use the Precision (P), Recall (R) and F_1-measure as the evaluation criteria. In addition, we add the Micro and Macro on multi-label classification (see Tables).

4.3 Experimental Results

4.3.1 Experiment with different features

As can be seen from Table 1, 2, and Table 3: the CNN_H method with DSE feature representation has the best performance among these models on all measures. But for the shallow learning (except CNN_H and CNN_flat) models, DSE doesn't do better than BOW+ performance because these models learn with the global feature and different documents in the same dimension is represented by the same word using BOW feature representation. But the DSE could not distinguish between different words in these shallow learning models and will bring extra noise.

Table 1: Precision (%) on dataset1 and dataset2.

Method	Dataset1		Dataset2	
	BOW+	DSE	BOW+	DSE
LDA	56.28	51.49	47.27	40.06
LR	54.54	50.97	45.81	38.11

[1] http://tinyurl.com/m2c8se6

[2] http://biotext.berkeley.edu/data.html

[3] http://www.bioasq.org/participate/challenges

[4] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

NB	55.70	47.05	46.79	40.68
SVM	58.02	50.64	48.74	40.96
PM	22.12	22.12	25.57	25.57
MTI	62.66	52.22	52.01	42.28
MU	62.69	50.56	51.41	39.54
CNN_flat	55.12	60.08	47.30	52.50
CNN_H	**61.39**	**68.76**	**51.54**	**61.33**

Table 2: Recall (%) on dataset1 and dataset2.

Method	Dataset1		Dataset2	
	BOW+	DSE	BOW+	DSE
LDA	55.89	50.98	46.94	39.66
LR	55.19	46.89	45.90	35.06
NB	55.31	46.63	46.46	40.28
SVM	57.61	50.43	49.71	40.59
PM	39.45	39.45	36.78	36.78
MTI	62.61	52.06	51.96	42.16
MU	62.10	50.10	52.90	39.19
CNN_flat	52.91	63.26	43.51	53.50
CNN_H	**63.17**	**72.54**	**52.00**	**65.01**

Table 3: F_1 (%) on dataset1 and dataset2.

Method	Dataset1		Dataset2	
	BOW+	DSE	BOW+	DSE
LDA	56.08	51.23	47.11	39.86
LR	54.86	48.85	45.86	36.53
NB	55.50	46.84	46.42	40.48
SVM	57.82	50.53	49.22	40.77
PM	28.35	28.35	30.17	30.17
MTI	62.63	52.14	51.99	42.22
MU	63.44	50.33	52.14	39.37
CNN_flat	53.99	61.63	45.32	52.90
CNN_H	**62.27**	**70.60**	**51.77**	**63.12**

While the DSE is not suitable for these shallow learning models, it does get good performances in the CNN model. There are several reasons for this results. Firstly this representation considers about the words sequence of the document and CNN's own strength is also more suitable to deal with the feature with sequence relationship, which is also the reason for the major breakthrough on speech in the NLP field. Although the word in the same dimension is not fixed, each word's 50-dimensional word embedding representation is different from other words' which can be distinguished well. And by setting the size of the sliding window and learning the local information between words, it shows a good way to suppress emergence of noise. Also CNN is trained by convolution, pooling and RELU, and these non-linear layers are also good solutions to solve the information interference problem brought from Unknown word when the document length is less than L. Secondly, the BOW representation is too sparse, especially when facing of unbalanced data, the difference between feature representations of documents is small. Category with a large number of samples will greatly interfere with the category with less samples, leading predictions to generally biased to high frequency label.

4.3.2 Experiment with different models

Table1, 2, 3 and 4 are all show hierarchical indexing performs much better than flat classification in processing large number of classes. Moreover, the coarse cluster step is an effective way to remove noise from the unbalanced distributed samples. Experimental statistics indicate that CNN_flat has a poor performance compared with CNN_H even they have the same DSE feature representation. By analyzing results from the test documents, we find that most test samples are predicted as negative samples by the model. Further checking the training stage, we find that there are very few class numbers for positive samples. Too many nodes are connected in the model and the adjustment of their weights should be updated in the fine-tunning process. A large number of weights are updated in negative samples than in positive samples which leads to this phenomenon.

Our proposed hierarchical indexing method (CNN_H) greatly increases the classification precision for the positive samples in the first layer. This hierarchical semantic framework can effectively reduce the impact of negative samples, and mainly update the weights connected with previous layer nodes during the next layer classification. The weight updates of nodes which are connected with different previous layer nodes are independent with each other, which greatly improve the effectiveness for training of positive samples.

Table 4: Classification results (%) on dataset3 (C=2000).

Method		MiP	MiR	MiF_1	MaP	MaR	MaF_1
BOW+	SVM	47.70	51.30	49.43	46.60	50.90	48.55
	MTI	59.74	54.32	56.90	56.57	48.51	52.23
	MU	59.70	54.28	56.86	56.68	48.17	52.08
	CNN_flat	54.81	48.20	51.29	44.50	47.92	47.92
	CNN_H	**60.01**	**53.90**	**56.79**	**61.80**	**51.40**	**56.12**
DSE	CNN_flat	59.32	53.94	56.50	56.18	48.51	52.06
	CNN_H	**67.51**	**61.38**	**64.30**	**63.93**	**62.57**	**63.24**

We also compare ReLU with the sigmoid function in our CNNs model. The network neurons with ReLU are reasonably sparse after training, thus gradients vanishing do not exist along with paths of active hidden units in an arbitrarily deep network. We also find that with “dropout” weight updating no longer depends on the interaction relationship among the hidden nodes. The “dropout” can provides a method to encourage each individual hidden unit to learn useful features without relying on other specific hidden units to correct its mistakes.

Significant test: Figure5 shows the F$_1$-measure performance comparison of the our method (CNN_H) against other approaches. From this figure, we observe the following: (1) The hierarchical indexing framework is better than the flat learning methods. (2) Our CNN_H method and CNN_flat get the first and second rank in all methods which show that based on CNN methods have better performance and more stable than shallow learning methods.

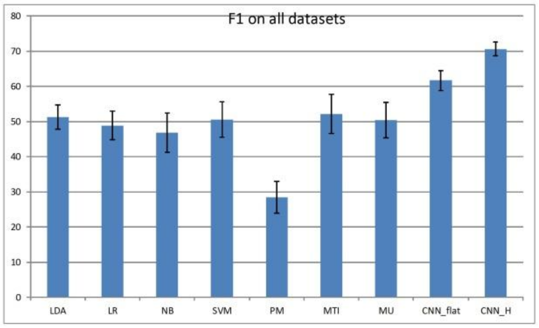

Fig. 5. Significant test on all dataset.

4.3.3 Experiment with DBN model

We also use DBN which we have described in related work to compare with CNN_H model in our paper. The figure6 shows the hierarchical classification method (CNN_H and DBN_H) better than the flat classification method (CNN_flat and DBN_flat). In addition, we also find that both DBN and CNN are deep learning methods, but CNN have better results than DBN in biomedicine abstract classification task. The difference is that CNN capture features by convolution calculated with sliding windows, then learning features wholly, but the DBN is to capture the overall characteristics directly. In this way of overall characteristics learning, DBN can solve two or multi-classes classification very well, but on the multi-label classification task, it's less effective than CNN model. This is because CNN model first gets the local features which can be considered as getting the key words of this local words. Then through the following convolution, pooling and fully-connection layers, it calculates the global characteristics of the whole document.

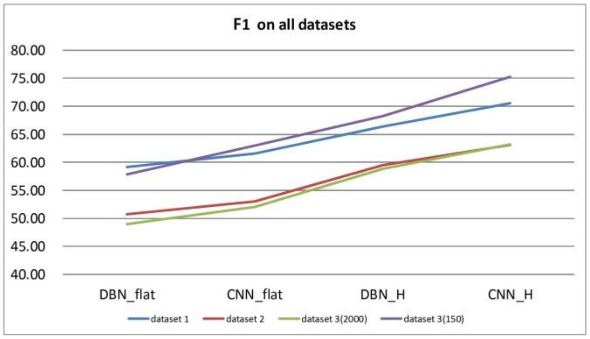

Fig. 6. Compared with DBN datasets.

5. CONCLUSION

In this paper, we propose a CNN based hierarchical model for document classification. By using tree-architecture unit to expand traditional classification method, experimental results show that our model has better effective performance than state-of-the-art methods. Besides, feature expanded by indexing is also more effective than BOW on abstract context. Because of the time limitation, this paper we only finish limited experiments and present related results in biomedical document indexing. Hence, in the near future we will continue to investigate and improve our proposed approach on a huge amount of semantic labels and a more massive biomedical documents.

REFERENCES

[1] Xue G R, Xing D, Yang Q, et al. Deep classification in large-scale text hierarchies[C]//Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2008: 619-626.

[2] Babbar R, Partalas I, Gaussier E, et al. On Flat versus Hierarchical Classification in Large-Scale Taxonomies[J].

[3] Dumais S, Chen H. Hierarchical classification of Web content[C]//Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2000: 256-263.

[4] Gopal S, Yang Y, Bai B, et al. Bayesian models for large-scale hierarchical classification[C]//Advances in Neural Information Processing Systems. 2012: 2411-2419.

[5] Gopal S, Yang Y, Niculescu-Mizil A. Regularization framework for large scale hierarchical classification[J]. 2012.

[6] Liu T Y, Yang Y, Wan H, et al. Support vector machines classification with a very large-scale taxonomy[J]. ACM SIGKDD Explorations Newsletter, 2005, 7(1): 36-43.

[7] Sun A, Lim E P. Hierarchical text classification and evaluation[C]//Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on. IEEE, 2001: 521-528.

[8] Cerri R, Barros R C, De Carvalho A C. Hierarchical multi-label classification using local neural networks[J]. Journal of Computer and System Sciences, 2014, 80(1): 39-56.

[9] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks[J]. Science, 2006, 313(5786): 504-507.

[10] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[11] Dos Santos C N, Gatti M. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts[C]//COLING. 2014: 69-78.

[12] Zeng D, Liu K, Lai S, et al. Relation Classification via Convolutional Deep Neural Network[C]//COLING. 2014: 2335-2344.

[13] Lin C, Zou Y, Qin J, et al. Hierarchical classification of protein folds using a novel ensemble classifier[J]. PloS one, 2013, 8(2): e56499.

[14] Hinton G E, Salakhutdinov R R. Replicated softmax: an undirected topic model[C]//Advances in neural information processing systems. 2009: 1607-1614. [18] A. Name, and B. Name, "Journal Paper Title", Journal Name, Vol. X, No. X, Year, pp. xxx-xxx.

[15] Phoungphol P, Zhang Y, Zhao Y. Robust multiclass classification for learning from imbalanced biomedical data[J]. Tsinghua Science and Technology, 2012, 17(6): 619-628.

[16] Yepes A J, MacKinlay A, Bedo J, et al. Deep belief networks and biomedical text categorisation[C]//Proceedings of the Twelfth Annual Workshop of the Australasia Language Technology Association. 2014: 123.

[17] Turian J, Ratinov L, Bengio Y. Word representations: a simple and general method for semi-supervised learning[C]//Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, 2010: 384-394.

[18] Aronson A R, Lang F M. An overview of MetaMap: historical perspective and recent advances[J]. Journal of the American Medical Informatics Association, 2010, 17(3): 229-236.

[19] Hu X, Zhang X, Lu C, et al. Exploiting Wikipedia as external knowledge for document clustering[C]//Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009: 389-396.

[20] Mikolov T, Yih W, Zweig G. Linguistic Regularities in Continuous Space Word Representations[C]//HLT-NAACL. 2013: 746-751.

[21] Aronson A R, Bodenreider O, Demner-Fushman D, et al. From indexing the biomedical literature to coding clinical text: experience with MTI and machine learning approaches[C]//Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing. Association for Computational Linguistics, 2007: 105-112.

[22] Fieschi M. Application of a medical text indexer to an online dermatology atlas[C]//Medinfo 2004. IOS Press, 2004, 107: 287.

[23] Trieschnigg D, Pezik P, Lee V, et al. MeSH Up: effective MeSH text classification for improved document retrieval[J]. Bioinformatics, 2009, 25(11): 1412-1418.

[24] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. the Journal of machine Learning research, 2003, 3: 993-1022.

[25] Patil A S, Pawar B V. Automated classification of web sites using Naive Bayesian algorithm[C]//Proceedings of the international multiconference of engineers and computer scientists. 2012, 1.

[26] Incorporating logistic regression to decision-theoretic rough sets for classifications.

[27] Chang C C, Lin C J. LIBSVM: a library for support vector machines[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2011, 2(3): 27