**深度学习技术在论文分类中的应用**

**刘晓东[[1]](#footnote-1)\*，倪浩然2，焦文彬**

信息化战略发展与评估中心，中国科学院计算机网络信息中心，北京 100190

摘要：【目的】我们使用深度学习模型对于文章多分类，研究论文发表机构的科研现状。【方法】我们设计了“多类别分类”模型，并应用卷积神经网络对中国科学院产生的8个不同主题的研究论文摘要进行分类。【结果】结果表明，科学研究涉及的学科交叉融合变得日趋紧密。【结论】多学科的融合交叉促进了科研产出，该研究可进一步用于科研机构的战略规划部署和评价等问题。

**Deep learning applications in paper classification**

Deep learning applications in paper classification

**Xiaodong Liu1\*, Haoran Ni2, Wenbin Jiao**

Center of Informatization Strategy and Evaluation, Computer Network Information Center Chinese Academy of Science, Beijing 100190, china

\*Email：liuxiaodong@cnic.cn

**Abstract**

**[Objective]** We use deep learning models to multi-classify articles. We use this method to analyze the situation of the corresponding institutions. [**Method]** In this paper, we designed an one-versus-rest category model and apply convolutional neural networks for classifying research papers. We conducted experiments on abstracts of research papers with 8 different subjects produced by Chinese Academy of the Sciences. **[Results]** The results show that the cross-integration of disciplines involved in scientific research is more and more common, and the integration of academic fields has also promoted the number of publications of scientific research papers. **[Conclusion]** **T**his research is conducive to the strategic planning and deployment of scientific research institutions, applied to the evaluation of discipline development, talents. Training and resource optimization issues.

Keywords**:** Text Classification, Natural Language Processing, Convolutional Neural Network, Classification algorithm

**Introduction/Overview**

The number of published papers contributed by the Chinese Academy of Sciences (CAS) is around 70000[[2]](#footnote-2) in the year 2018, which rose by 21.8% year-on-year. Due to the increasing number of papers each year and the disciplinary inter-crossing, artificial classification of papers by disciplines becomes harder and harder. There are several difficulties with the artificial classification. First, research paper contains massive information, which makes the workload much heavier than other classification tasks. Secondly, reading papers from specific disciplines requires professional knowledge. However, hiring experts for the task is expensive and impractical. Finally, the inter-crossing of disciplines is hard to tell even when the papers are evaluated by experts in the related fields. In the meantime, the demand of classifying papers and trend prediction is rapidly increasing. In this background, text classification in natural language processing (NLP) is considered to solve this problem.

As an essential component in NLP applications, text classification has aroused the interest of many researchers. There are many classic application scenarios for it, such as public opinion monitoring (Bing & Lei, 2012)[1], intelligent recommendation system, information filtering and sentiment analysis (Aggarwal & Zhai, 2012)[2].

One of the main problems in text classification is the word representation. Bag-of-Words (BoW) model (Zhang et al., 2010)[3] is the pioneer in this area, where some designed patterns, such as uni-grams, bi-grams and n-grams are extracted as features. With the higher order in n-grams and the complicated feature structure (Post & Bergsma, 2013)[4], the model could include more contextual information and word orders. However, it failed in capturing the semantics of the words and the data sparsity problem remains unsolved. As a distributed representation of words, pre-trained word embedding offers a way to capture meaningful syntactic and semantic regularities (Mikolov, Chen et al., 2013)[5], and lesser the data sparsity problem (Bengio et al.,2003)[6]. The state-of-art method of word embedding is word2vec (Rong, 2014)[7], which includes two training models: continuous Bag-of-Word (CBoW) (Mikolov, Chen et al., 2013)[5] and skip-grams (Mikolov, Sutskever et al., 2013)[8]. In this paper, we apply the Global Vectors for Word Representation (GloVe) model proposed by Pennington et al. (2014)[9], which is proved to outperform word2vec, on the word representation task.

Another problem is the classifier. There are many machine-learning algorithms that can be used as classifiers, such as logistic regression (LR), naive Bayes (NB) and support vector machine (SVM). However, these methods all have the data sparsity problem and do not have a satisfying performance on NLP tasks. With the development of deep neural networks, convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are widely used in NLP tasks as the state-of-art classifiers due to their exceptional performance. In our research, we build a CNN (Kalchbrenner et al., 2014)[10] model and apply it to the paper classification. The input of the model handles sentences of varying length. The layers include several one-dimensional convolutional layers and a *k*-max pooling layer. The convolutional layers apply one-dimensional filters across each row of features. The max pooling layer is a non-linear sub-sampling function that returns the maximum of a set of values. A convolutional layer followed by a max pooling layer and a non-linearity form a feature map. In the input layer, we enrich the representation by computing multiple feature maps with disparate filters. Subsequent layers followed by the input layer also have multiple feature maps computed by convolving filters with all the maps. The weights at these layers form an order-4 tensor. The resulting structure is named as Convolutional Neural Network (Lecun et al., 1998)[11].

The networks are trained on 7000 abstracts of the research papers from 23 different subjects labelled artificially by experts in CAS. With the auto-optimization of hyper-parameters, the networks achieve generally greater than 90% in the prediction accuracy on the hand-labelled test. We experiment with the well-trained networks in the published research papers contributed by the CAS from 2012 to 2018. By labelling the papers with the classifiers of selected subjects, the trend of interdisciplinary research is shown in the results.

The outline of the paper is as follows. Section 2 reviews the related works including word representation models, neural networks, optimization of hyper-parameters and related applications of text classification. Section 3 defines the classifier, word representation model, the relevant operators and the layers of CNN and the auto-optimization method we applied in the paper. Section 4 discusses the experiments and the results.

**1. Related Works**

The text classification tasks mainly include three parts: feature engineering, feature selection and classifiers. The feature engineering was commonly based on the BoW model. Rong (2014)[7] proposed the word2vec method, which is widely accepted as the state-of-the-art for word representation. However, it is limited in utilizing statistical information, since there is no global co-occurrence counts. Pennington et al. (2014)[9] introduced the GloVe model, which combines the advantages of latent semantic analysis (LSA) (Deerwrester et al., 1990)[12] and word2vec. This new word-representation method efficiently improved the accuracy of many basic NLP tasks, especially for limited training dataset. There are also other advanced methods recently, such as Bidirectional Encoder Representation from Transformers (BERT1) (Devlin et al., 2018)[13]. Feature selection is applied for filtering the noisy features and improving the performance of the classifiers. The common feature selection methods include removing stopping words, statistic, mutual information and *L1* regularization (Ng, 2004)[14]. For classifiers, there are several machine learning algorithms such as logistic regression (LR), naive bayes (NB), and support vector machine (SVM). However, they all have the data sparsity problem.

**1.1 Neural Networks in NLP**

With the development of deep neural networks, the data sparsity problem has been well solved. Many neural models have been proposed for word embedding and classification. Recursive Neural network (RecursiveNN) (Socher, Huang et al., 2011; Socher, Pennington et al., 2011; Socher et al., 2013)[15][16][17] captures the semantics of a sentence via a tree structure. However, the time complexity of constructing a textual tree is at least *O()*, which is too time-consuming for long documents. The Recurrent Neural Network (RNN) (Elman, 1990)[18] is a model, which analyzes the semantics of the text in fixed-sized hidden layers, with a time complexity *O(n)*. However, it is a biased model where later contents are more dominant than earlier contents. To tackle the bias, the Long Short-Term Memory (LSTM) (Liu et al., 2016)[19] model is proposed, which gives a better performance for long documents. The Convolutional Neural Network (CNN) (Kalchbrenner et al., 2014)[10] is another unbiased model introduced to NLP tasks. Although the time complexity of CNN is *O(n)* too, we found that empirically it outperforms LSTM in both training time and accuracy. Advanced models include Attention Networks (Abreu et al., 2019)[20], Contextual LSTM (C-LSTM) (Ghosh et al., 2016)[21] and Recurrent CNN (RCNN) (Lai et al., 2015)[22]. We will not discuss them since they are too sophisticated and do not contribute noteworthy improvement in accuracy.

The performance of the model also depends on the optimizer and the hyper-parameters. There are many classic optimizers, such as Batch Gradient Descent (BGD), Stochastic Gradient Descent (SGD), Momentum, Adadelta, Adam. We apply the SGD method in our experiments due to its sensitivity to hyper-parameters and model performance. Zhang and Wallace (2015)[23] researches deeply in the sensitivity analysis and gives some practical advises in the selection of hyper-parameters. However, it requires rich experience in NLP engineering, and consumes massive energy and time. Instead of manual selection, we apply auto-optimization algorithms in our research, which offers a modest performance with time and energy saving. The most direct algorithm is Grid Search. However, it costs too much computing resources. Bergstra and Bengio (2012)[24] proposes Random Search which improves the performance of optimization. Bayesian algorithm (Snoek et al., 2012)[25] is more efficient than Random Search, but it cannot deal with continuous variables. To tackle this problem, we apply Tree-structured Parzen Estimator (Bergstra et al., 2011; Bergstra et al., 2013)[26][27] for optimizing the hyper-parameters. To exploit the parallelism of the operations, the network is trained on GPUs using Pytorch.

**2. Methods**

**2.1 Classifier**

For training sample sets {}, where ,n=1,2,3,4 is the sample set with the category, we define the classifier as an one-versus-rest model in our task.

Training Samples

+ - Classifier Prediction

Final Result

{[], []} => -

{[], []} => -

{[], []} => +

{[], [] => -

In our research, we train separate classifiers for different subjects and label each paper with these classifiers according to the above model. If there is more than one positive prediction for the paper, we believe cross-discipline happens between the labelled subjects.

**2.2 Word Representation Learning**

We introduce GloVe model in this section. The model is proposed by Pennington et al. (2014)[9]. We denote the matrix of word-word co-occurrence counts as ***X***, where the entries tabulate the number of times word *j* occurs in the context of word *i*. The number of times any word appears in the context of word *i* is denoted as . We denoteas word vectors and as separate context word vectors. The model is given as follows:

 **(1)**

where *V* is the size of the vocabulary, and are the bias. We practically choose the function:

 **(2)**

as the weighting function . We fix and to give a best performance for our experiments (Mikolov, Chen et al., 2013)[5].

**2.3 Convolutional Neural Network**

We build a model using a convolutional architecture that alternates 3 or 4 convolutional layers with pooling layers given by *k*-max pooling. The resulting architecture is the Convolutional Neural Network.

**2.3.1 Convolution**

For a vector of weights  and a vector of inputs  as the sentence, the one-dimensional convolutional operation is to take the dot product of and each *m*-gram of  to obtain the vector :

 **(3)**

There are two types of convolution depending on different requirements. The narrow type requires  and the resulting sequence , where *j* is from *m* to *s*. In the wide type of convolution, *s* could be smaller than *m*. the input  are taken to be zero when *i<1* or *i>s*. In the case, it yields the sequence , where the range of the index *j* is 1 to *s+m-*1. Notice that, the sequence  in the narrow convolution is a sub-sequence of which in the wide convolution. Wide convolution has some advantages such as ensuring that all weights in the filter reach the entire sentence and producing a non-empty  which are independent of the width *m* and the sentence length *s* all the time. We apply wide convolution operation in our model. For the sentence matrix  which is constructed by the embedding for each word, we obtain a convolutional layer by convolving a matrix with trained weights with the activations at the layer below. The resulting matrix  at each layer has dimensions .

**2.3.2 Non-Linear Activation Function**

After each convolutional layers, a bias  and a non-linear activation function *g* are applied component-wise to the resulting matrix  of the convolution. Define *M* to be the matrix of diagonals:

 **(4)**

where are the weights of the *d* filters of the wide convolution. Thus, we obtain:

 **(5)**

This function makes sure that the output of our network is not just another linear combination of the inputs and gives the possibility to generate more complicated functional relationship for different tasks. There are many different activation functions, such as *Sigmoid*, *tanh* (Gulcehre et al., 2016; Glorot & Bengio, 2010)[28][29], *Relu* (Nair & Hinton, 2010; Glorot et al., 2010)[30][31] and *maxout* (Goodfellow et al., 2010)[32]. We apply *Relu* function due to its faster convergence rate and generally moderate performance in practice.

**2.3.3 Max Pooling**

Given a number *k* and a sequence , where , *k*-max pooling is the pooling operation that extracts the *k* largest values of with the same order and constructs a sub-sequence . In our experiments, we set *k=1* in the *k*-max pooling layers.

 **(6)**

where the *j*-th sequence of the matrix  is the maximum in the *j*-th elements of .

By applying the *k*-max pooling operators after the convolutional layers, it is available to pool the *k* most active features in and to capture the information throughout the entire text. It preserves the order of the features and ignores their positions. It also discerns precisely the number of times the feature is highly activated in and the change of the feature activations across . There are other kind of pooling operators such as average pooling (Collobert et al., 2011)[33]. We do not apply them here since only a few words are meaningful for capturing the semantics and further classification in the text. After all the convolutional layers and max pooling layers, a fully connected layer followed by the softmax function is applied for converting the output numbers into probabilities, which predicts the classes of the input sentence.

 **(7)**

For binary classifier, we let *n=2* in both Equation 6 and Equation 7.

**3** **Experiments**

**3.1 Datasets**

Our datasets are collected from the inner database of CAS. The datasets include 330907 published papers of CAS from 2012 to 2018. We randomly take 7000 abstracts out of the papers as the training samples, and label them with 23 subjects by experts. In the word representation training, our pre-trained vectors for GloVe are 50-dimensional *Wikipedia* 2014 + *Gigaword* 5, which are provided in the GloVe project (Pennington et al., 2014)[9]. After that, we classify the remaining papers by 8 main subjects, which are Astronomy, Atmospheric Science, Electronic Science and Technology, Geology, Mathematics, Nuclear Science and Technology, Physics and Computer Science and Technology, using the well-trained models. We investigate the cross-disciplines between Computer Science and Technology and other 7 selected subjects on timeline.

**3.2 Hyper-Parameters and Training**

In our experiments, the optimization goal is to minimize the difference between the prediction and the true distribution, which includes an  regularization term over the parameters. The set of parameters in the optimization are the word embeddings, the filter and the fully connected weights. The network is trained with mini-batches by back-propagation. Stochastic gradient descent is applied here to optimize the training target. The hyper-parameters comprise the batch size, the learning rate of optimization, the number of negative samples, the number of layers, the kernel number, and the kernel size of each convolutional layer. To optimize these hyper-parameters, the Tree-structured Parzen Estimator Method is applied here as an auto-optimization algorithm.

**3.3 Experiment Settings**

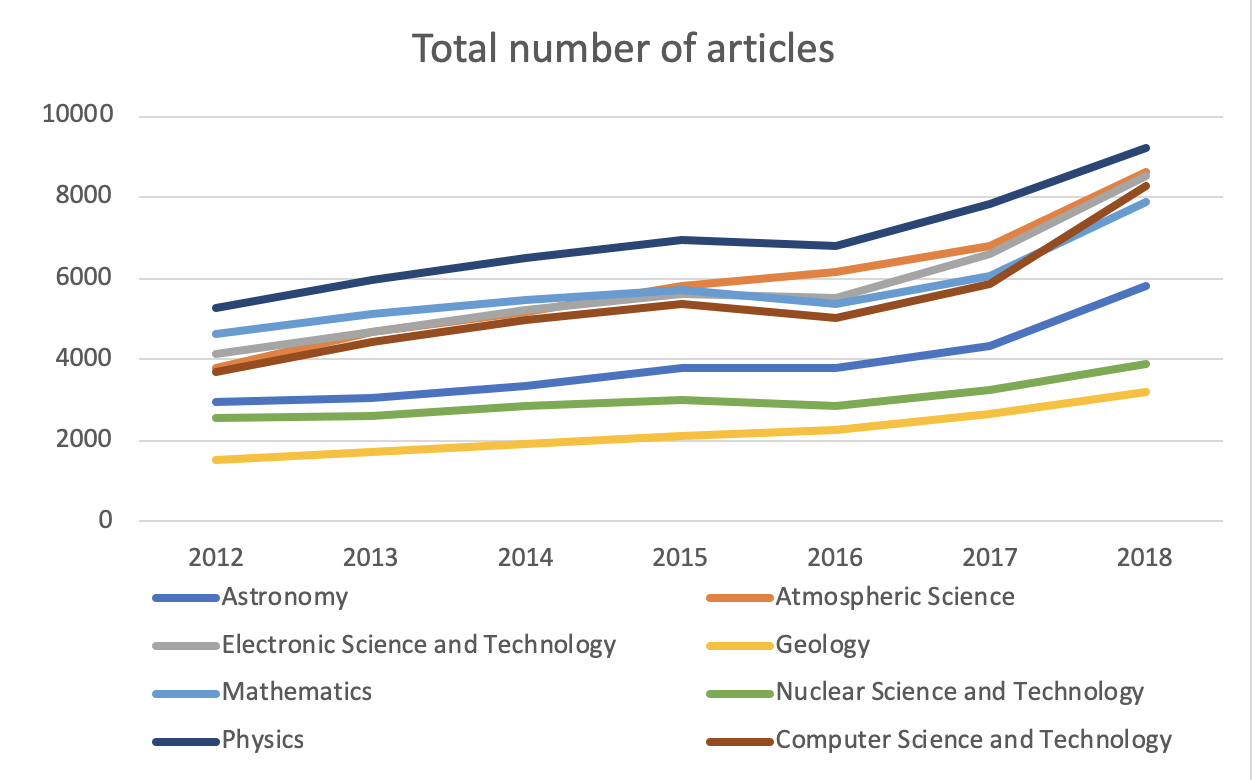
We preprocess the datasets as follows. We use Natural Language Toolkit (NLTK) to obtain tokens and stems. Stop words and symbols are also removed by this toolkit. We split each dataset into training dataset, validation dataset and testing dataset by the ratio 6:2:2. We apply F1 measure to evaluate the model performance, Precision, Recall and Accuracy to evaluate the classification results. The hyper-parameter settings of the neural networks depend on the datasets being used.

We use the CNN model to classify the abstract information of the paper, and use 3 or 4 layers of convolutions to train the observation effect. The algorithm of the training model chooses to use the stochastic gradient descent algorithm. Because the stochastic gradient descent algorithm is more sensitive to hyperparameters. and get better results by adjusting the parameters.

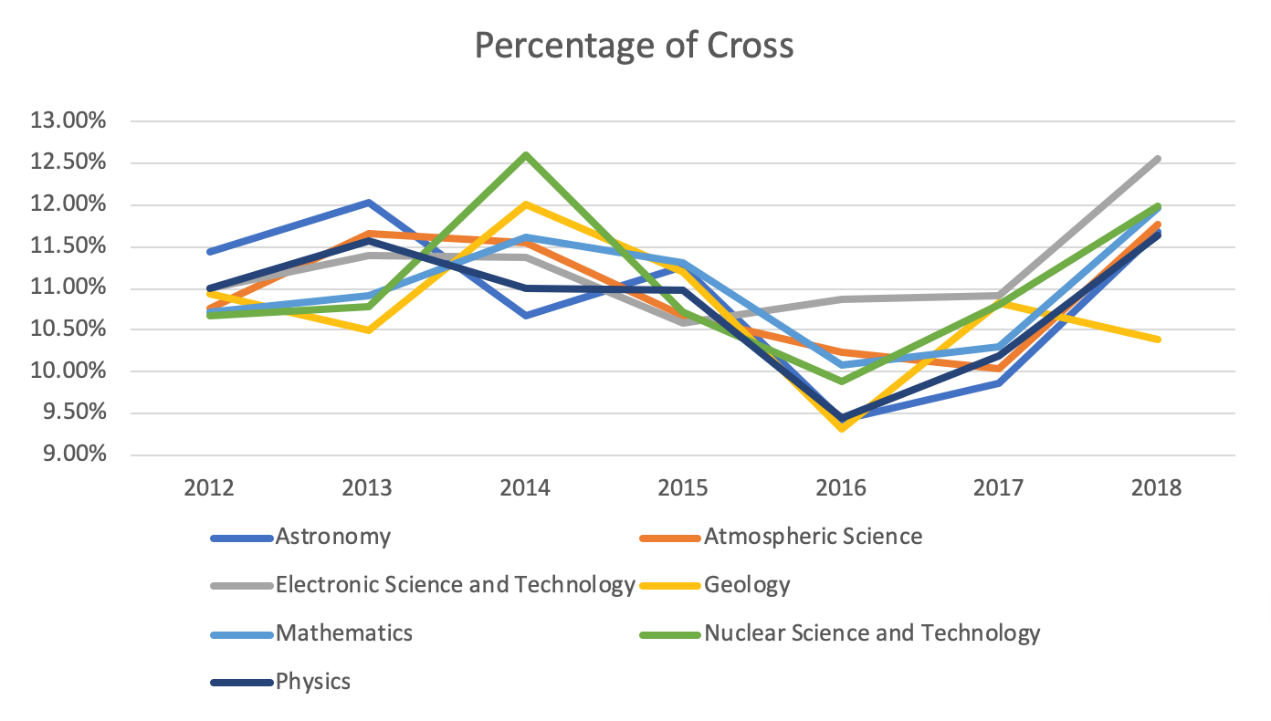
**3.4 Results and Discussion**

**3.4.1 Interdisciplinary Design**

According to the experimental research in this paper, due to the rapid development of computer science Informaionization, the development of information technology represented by cloud computing, big data, deep learning, etc. has entered an explosive phase since around 2010, and a large number of cutting-edge technologies of computer science have been used for each Scientific research in the field. As shown in the Figure 1, the field of physics ranks first in the number of 48,450 papers published in 6 years. Using the method proposed in 4.1.2 of this paper, 5,254 papers in the field of physics are used to apply the knowledge in the field of computer science and technology. The crossover rate is as high as 11.1%, ahead of the other seven areas of the dataset. In Figure 2, the integration of various fields and computer science and technology is getting closer and closer, and the correlation between various scientific research work and computer science and technology is getting stronger and stronger.



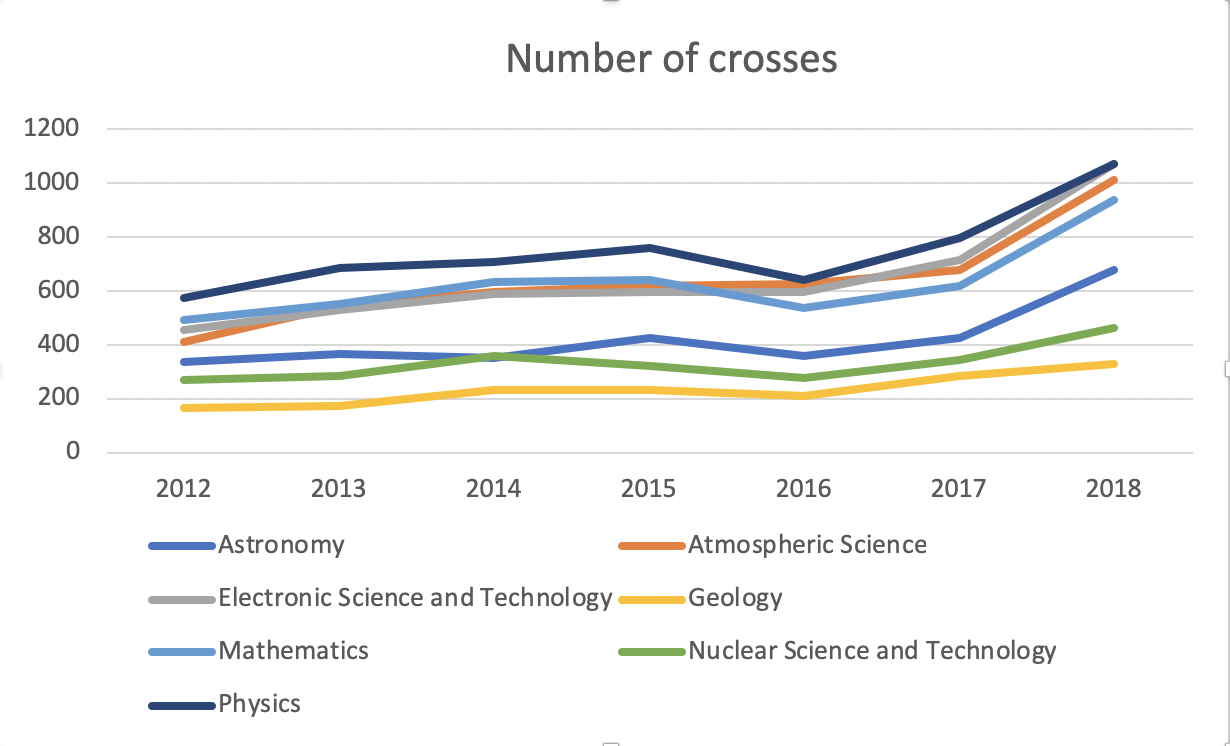
**Figure 1: Total number of Articles**



**Figure 2. Percentage of Cross**

**3.4.2 Disciplinary development and interdisciplinary**

Secondly, in recent years, in the field of physics, in recent years, distributed file systems, machine learning, cloud storage and other technologies have been widely used in the field of physics. The number of papers published in this field has increased from 5,252 in 2012 to 9204 in 2018, the number of interdisciplinary papers in the field has also increased from 578 to 1072, both of which have shown a consistent trend of overall curve rise; the same is true in the field of atmospheric science, which grew from 3,808 to 8598 articles, the cross-disciplinary relationship between this field and computer science and technology has also increased from 410 to 1011. The growth rate is obvious. The focus of electronic science and technology field is from 4152 in 2012 to 8531 in 2018, and it is related to computer science. The interdisciplinary division of technology has also grown from 457 to 1,072. the overall focus is on the whole. Mathematics, astronomy, and other fields are all growing as shown in Figure 3.



**Figure 3: Number of Crosses**

**3.4.3 Research Ability and Subject Cross**

In the end, according to our findings, the degree of information infrastructure construction of scientific research institutions shows that the degree of information technology infrastructure construction of scientific research institutions has been steadily increasing year by year, and the scientific research capabilities of large scientific research institutions have been greatly improved. It is consistent with the trend of integration of various research fields with computer science and technology, according to which we believe that there is a positive correlation between the ability of scientific research and the degree of interdisciplinary. The scientific research ability and the degree of interdisciplinary mutual promotion and growth.

**3.4.4 Research Extension**

For the subject classification and fusion problem of this paper, we can apply the results to the evaluation of subject development, talent cultivation, resource optimization, and the establishment of college education training mode curriculum, or the cultivation of compound talents. The research results in this paper can better guide scientific research institutions to carry out strategic planning and deployment. In addition, our research ideas can also be applied to related expansion issues, such as subject type prediction, attribute judgment and differentiation of low-frequency subjects and confusing subjects, introduction of attention mechanism training attribute-based subject prediction models, and, for example, criminal case records. The problem of classification and prediction of complex case types, based on the case description of the case, predicts the final judgment result.

**CONCLUSION & DISCUSSION**

In this paper, we applied Convolutional Neural Network and Recurrent Neural Network in Research Paper Classification Tasks. We researched the published papers of research institutions over the years, marked the papers and selected eight types of them to train, and used the algorithm model of 1VO+GloVE+CNN to preprocess and train the dataset. The classification and interdisciplinary statistical work based on these eight subject models were completed to analyze the number, proportion and regularity of papers with interdisciplinary subjects. Through experiments, this paper believes that the intersection of scientific research work and computer science in various fields is becoming more and more close, and the degree of interdisciplinary degree is consistent with the growth trend of the number of scientific papers published. In the future work, we can apply scientific research results to the development of disciplines, personnel training and resource optimization, which will help scientific research institutions guide scientific research institutions to carry out strategic planning and deployment. In the future, we will continue to study different areas on this issue.

**Acknowledgments**

This work was supported by the National Science Library of the Chinese Academy of Sciences. Thanks to the National Science Library.

**References**

1. Bing L, & Lei Z. (2012). A survey of opinion mining and sentiment analysis.
2. Aggarwal C, & Zhai C. A survey of text classification algorithms, 2012.
3. Zhang Y, Jin R & Zhou Z., Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning & Cybernetics* (2010): 1(1-4), 43-52.
4. Post M & Bergsma S., Explicit and implicit syntactic features for text classification. *In Proceedings of the 51st annual meeting of the association for computational linguistics* (2013): volume 2: Short papers, 866–872.
5. Mikolov T, Chen K & Corrado G. et al., *Efficient estimation of word representations in vector space*, 2013.
6. Bengio Y, Ducharme R & Vincent P et al., A neural probabilistic language model. *Journal of machine learning research* (2003): 1137–1155.
7. Rong X., word2vec parameter learning explained. *Computer Science* (2014).
8. Mikolov T, Sutskever I & Chen K. et al., Distributed representations of words and phrases and their compositionality, 2013.
9. Pennington J, Socher R & Manning C. D., *Empirical methods in natural language processing* (2014): 1532–1543. Retrieved from http:// www.aclweb.org/anthology/D14-1162
10. Kalchbrenner, N., Grefenstette, E., & Blunsom, P., A convolutional neural network for modelling sentences, 2014.
11. Lecun Y, Bottou L & Bengio Y et al., Gradient-based learning applied to document recognition.Proceedings of the IEEE (1998): 86(11), 2278-2324. doi: 10.1109/5.726791.
12. Deerwester S, T. Dumais S, & Landauer T et al., Indexing by latent semantic analysis. *Journal of the American Society for Information Science* (1990): 41, 391-407. doi:10.1002/(sici)1097- 4571(199009)41:660;391::aid-asi162;3.0.co;2-9.
13. Devlin J, Chang M, Lee K et al., Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.
14. Ng A. Y., Feature selection, l1 vs. l2 regularization, 2004.
15. Socher R, Huang E. & Pennington J. et al., Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. Advances in neural information processing systems (2011): 24.
16. Socher R, Pennington J & Huang E. et al., Semi-supervised recursive autoencoders for predicting sentiment distributions.In Conference on empirical methods in natural language processing, 2011.
17. Socher R, Perelygin A & Wu J. et al., Recursive deep models for semantic compositionality over a sentiment treebank, 2013.
18. Elman J. L., Finding structure in time. *Cognitive Science* (1990): 14(2), 179-211. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1402\_1doi:10.1207/s1551 6709cog1402\\_1.
19. Liu P, Qiu X & Huang X., Recurrent neural network for text classification with multi-task learning, 2016.
20. Abreu J, Fred L & Macêdo D et al., Hierarchical attentional hybrid neural networks for document classification, 2019.
21. Ghosh S, Vinyals O, Strope B et al., Contextual lstm (clstm) models for large scale nlp tasks, 2016.
22. Lai S, Xu L & Liu K et al., Recurrent convolutional neural networks for text classification. In In proc. conference of the association for the advancement of artificial intelligence, 2015.
23. Zhang Y, & Wallace B., A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification*. Computer Science* (2015).
24. Bergstra J, & Bengio Y., Random search for hyperparameter optimization. *Journal of Machine Learning Research* (2012): 13(1), 281-305.
25. Snoek J, Larochelle H & Adams R. P., Practical bayesian optimization of machine learning algorithms. In International conference on neural information processing systems, 2012.
26. Bergstra J, Bardenet R & Bengio Y et al., Algorithms for hyper-parameter optimization.In International conference on neural information processing systems, 2011.
27. Bergstra J, Yamins D, & Cox D., Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. *Journal of Machine Learning Research* (2013): 115-123.
28. Gulcehre C, Moczulski M & Denil M. et al., Noisy activation functions*,* 2016.
29. Glorot X & Bengio Y. Understanding the difficulty of training deep feedforward neural networks. *Journal of Machine Learning Research* (2010): 9, 249-256.
30. Nair V & Hinton G. E., Rectified linear units improve restricted boltzmann machines. In International conference on international conference on machine learning, 2010.
31. Glorot X, Bordes A & Bengio Y., Deep sparse rectifier neural networks, *Journal of Machine Learning Research* (2010): 15.
32. Goodfellow J, Warde-Farley D & Mirza M et al., Maxout networks. *30th International Conference of Machine learning* (2013): (3), Vol.28, 1319-1327. Retrieved from http://dblp.uni-trier.de/db/conf/ icml/icml2013.html#GoodfellowWMCB13.
33. Collobert R, Weston J, & Bottou L et al., Natural language processing (almost) from scratch. *Journal of Machine Learning Research* (2011): 12(1), 2493-2537.

1. [↑](#footnote-ref-1)
2. The data in this paper is collected from the public database of

   Web of Science. [↑](#footnote-ref-2)