Scientometrics (2020) 123:813–839
https://doi.org/10.1007/s11192-020-03385-w

Research on classification and similarity of patent citation based on deep learning

Yonghe Lu[1] · Xin Xiong[1] · Weiting Zhang[1] · Jiaxin Liu[1] · Ruijie Zhao[1]

Received: 4 August 2019 / Published online: 28 February 2020

Abstract
This paper proposes a patent citation classification model based on deep learning, and collects the patent datasets in text analysis and communication area from Google patent database to evaluate the classification effect of the model. At the same time, considering the technical relevance between the examiners' citations and the pending patent, this paper proposes a hypothesis to take the output value of the model as the technology similarity of two patents. The rationality of the hypothesis is verified from the perspective of machine statistics and manual spot check. The experimental results show that the model effect based on deep learning proposed in this paper is significantly better than the traditional text representation and classification method, while having higher robustness than the method combining Doc2vec and traditional classification technology. In addition, we compare between the proposed method based on deep learning and the traditional similarity method by a triple verification. It shows that the proposed method is more accurate in calculating technology similarity of patents. And the results of manual sampling show that it is reasonable to use the output value of the proposed model to represent the technology similarity of patents.

Keywords Deep learning · Patent citation · Classification · Technology similarity

✉ Yonghe Lu
luyonghe@mail.sysu.edu.cn

Xin Xiong
xiongx33@mail2.sysu.edu.cn

Weiting Zhang
874120193@qq.com

Jiaxin Liu
liujx28@mail2.sysu.edu.cn

Ruijie Zhao
zhaorj8@mail2.sysu.edu.cn

[1] School of Information Management, Sun Yat-sen University, Guangzhou, China

Introduction

Patents, as a type of scientific literatures, have many different characteristics from papers. In terms of document citation, papers tend to cite a large number of related literatures in order to share and verify research results. There is not necessarily a direct knowledge association between citing and cited references (Gilbert 1977). However, patent applicants are unwilling to cite other patents due to their competitive relationship with other patents, or they cite other patents in the form of rebuttal in order to emphasize the advantages of the technologies of pending patents (Li et al. 2012). Unlike papers, patent citations come not only from applicants but also mainly from patent examiners. In order to verify innovation of patents, patent examiners tend to cite as many comparative patents related to the patented technology as possible (Rui and Liansheng 2009). Therefore, these differences should be taken into account in the analysis of patent citations (Jaffe et al. 1993), and the analysis methods cannot be the same as paper. However, most of the existing researches do not pay attention to the differences, which may bring new opportunities. Focused on the characteristic that the citations of examiners have definite technical relevance to the pending patents, this study regards citation relationship as the label of technical relevance to conduct the calculation and forecast of technology similarity between two patents, which expanded the subject of patent research.

During the 12th five-year plan period, various regions implemented the intellectual property incentive policies such as preferential patent maintenance fee and tax treatment for enterprises, which promoted the innovation-driven development of the country. As a result, compared with the end of the 11th five-year plan period, the number of invention patents per 10,000 population reached 6.3, increased of three times. The rapid growth of the number of patents has not only increased the number of examiners who review them, but has also continued to increase the amount of time and effort they spend on searching for "comparison documents". Therefore, if technical means can be adopted to automatically recommend "comparison documents" based on the pending patent, the examination efficiency can be improved. In addition, the automatic recommendation method can also help the applicants. Since the applicants must provide enough comparative documents just like the patent examiners according to the U.S. patent law. If violating this regulation, the applicants may lose the authorization. Therefore, using technical means to provide "comparison documents" automatically according to the pending patent for the examination can not only save time and manpower, but also avoid the illegal problems caused by poor retrieval ability of applicants. More broadly, the technology can also be extended to the field of paper writing and manuscript review after further improvement, making the writing and retrieval process more rigorous.

Nowadays a large number of achievements has emerged in various fields using deep learning methods. However, no matter processing texts , pictures, audios, videos and other different data formats, or using CNN, RNN and other different neural network models, most of the deep learning methods are always used to solve the problems of classification and prediction. It is difficult to use deep learning to calculate the amount of information contained in a text as well as judge the strength of the relationship between two objects. In addition, the existing text similarity methods calculate the similarity of documents based on the distance between two text vectors. Text similarity generally determine the patents with similar contents and topics, which is insufficient when determining whether the adopted technology of two patents are similar. It usually uses statistical or manual filtering methods to extract relevant features, which cost a lot of manpower and fail to consider

the semantic and contextual relations. Because deep learning is more intelligent and can automatically learn the semantic relationships in the texts, we combine this method with the characteristics of patent citations. *Technology similarity* means the patents with similar technology to solve different problems, which can be reflected in its examiner citation patent. It means one patent has high technology similarity with its examiner citation patent, not just text similarity. But if one patent has text similarity with another in topic, it may not have technology similarity with another. This study proposed to incorporate patent citation into binary classification calculation and use the probability output by the patent citation classification model to represent technology similarity of two patents, which extends deep learning methods and the significance of text similarity.

This paper proposes a patent citation classification model based on deep learning. RNN considers the characteristics of temporal information so that it can encode context information in the text mining process. While CNN can reduce the complex matrix to a vector, which benefits the image and text mining. This paper considers using RNN to encode the context and CNN to reduce dimensions of the encoded matrix to a vector for document representation. Finally the model realizes the classification of patent citation relationship. In addition, the deep learning model generally outputs multiple continuous values through SoftMax classifier in the last layer, and selects the category of the maximum value as the classification result. In this paper, the output continuous value is assumed as the value to determine the technology similarity between two patents. We apply statistics method and manual spot check to illustrate the rationality of the hypothesis.

The main contributions of this paper include crawling two patent datasets on different fields, experimenting on combinations of different traditional or deep learning models to verify correctness and superiority of proposed patent citation classification model. In addition, this paper proposes to use the output value of the proposed model to represent the technology similarity of patents. Through statistics and manual methods, this paper demonstrates its rationality and superiority to other traditional similarity methods. This paper consists of six parts. The first part explains the research background of this paper. The second part introduces relevant theories and techniques. The third part proposes the methodology of patent citation classification and technology similarity based on deep learning. The fourth part takes the patent data in the field of text analysis and telecommunication technology as an example to realize the construction of the deep learning model proposed in this paper. The fifth part is the discussion of classification result and the patent technology similarity. The sixth part is the conclusion.

Related works

Semantic matching

Semantic matching is a technology in computer science to identify semantic related information. At present, the most common semantic matching applications in the field of deep learning are Question Answering System (QA), Semantic Relatedness and Natural Language Inference (NLI).

Question answering system The task of question answering system is to select one or several results from a large number of alternative answers on the basis of a given query, which is one of the key technologies in the field of information retrieval. At present, the most

important semantic matching model in the field of information retrieval is DSSM (Huang et al., 2013) and its variants. The input of DSSM for Chinese and English is processed with word hashing and one-hot vector respectively, while the BOW method which ignores the word order is selected for text representation. Finally in the matching layer, the cosine distance is calculated to determine the semantic similarity between the query and the document. Shen et al. (2014) proposed CNN–DSSM as a further improvement of DSSM. CNN–DSSM replaces the BOW and DNN of the representation layer with CNN to capture the global context features in the text. LSTM–DSSM Palangi et al. (2014) is another improvement of DSSM which also replaces the BOW and DNN of the representation layer but with LSTM. As LSTM can solve the problem of long-term dependence, the improved method can better consider the semantic information of the contexts.

Semantic relatedness Semantic relatedness refers to the similarity or relevance of two texts or sentences at the semantic level. At present, researches are more focused on how to identify specific semantic relations or select the right matching text, namely NLI and QA. There are few researches using deep learning method to output the similarity value between text pairs. Tai et al. (2015) introduced the tree-based LSTM model which is different from the traditional chain LSTM and can better reveal the syntactic information for text representation. In the task to calculate the semantic relatedness, the model obtains a probability vector of five categories through Softmax classifier. The labels of the categories are the similarity scores ranged from 1 to 5. The predicted score is not the score corresponding to the category with the largest probability, but the weighted sum of the category scores and the predicted probability vectors, which ensures the continuity of the relatedness score. On the basis of Tai, Zhou et al. (2016) used sequence RNN to obtain the sentence representations, and then input the representations into tree-based RNN to encode the dependency structures in the sentences. At the same time, the attention mechanism was introduced to add weights to the hidden states that represent the words in the sentence. In this paper, the prediction of the patent citation relationship is a binary classification problem. Meanwhile, since the existence of citation relationship is related to the technology similarity between the patent pairs, the probability value of the existence of citation relationship output from SoftMax is considered as the technology similarity of the patent pairs, which is consistent with the semantic relatedness score defined by Tai.

Natural language inference The main task of Natural Language Inference (NLI) is to judge the semantic relationship between two sentences. One of the most important problems is to recognize the semantic entailment. Entailment refers to the implication relation in first-order logic, which takes one text as the premise and the other text as the hypothesis. If the hypothesis can be inferred from the premise, then it's considered that the two texts exists entailment relationship. Yin et al. (2016) introduced three methods of using attention mechanism in the CNN. The first method is to calculate the attention matrix between two sentences before convolution which is input into the convolution process together with the feature matrices of the two sentence. The second one is to calculate the attention matrix according to the results of the convolution before pooling. The last one is the combination of the first and the second methods.

Based on the above discussion, it can be seen that the models used in these three semantic matching tasks are very similar. In general, the semantic matching model has shown the following development rules in recent years:

(1) From representation-based matching model to interaction-based matching model. The representation-based matching model is a method to encode the text semantically and

then calculate the matching score. This method ignores the influence of the position and order of the words in the text pair on the matching result. But the interaction-based matching model is a method that firstly generates the matching signal matrix between words by matching text pairs, and then obtains the matching results through subsequent encoding. This method remains the position and order information of words in the text, and has gradually become the mainstream of the semantic matching model after 2016 because of the better experimental effect (Zhang et al. 2017; Conneau et al. 2017).

(2) From simple encoding to deep and multi-granularity encoding. The BOW text representation method adopted earlier is simple to operate, but ignores the information of word order. Then RNN and CNN are introduced to encode contextual information and highlight local features of the text and the attention mechanism is used to recommend the importance of the words and sentences in the text. Furthermore, the depth and granularity of the model is increased to reveal semantic information in different aspects of text. Semantic matching model is constantly transforming to a more complex and intelligent direction.

(3) From a single model to multiple model combination. In the early semantic matching models, only one of the DNN, CNN or RNN models was selected to encode and represent the text, and the effect was improved by increasing the complexity of the model and introducing the attention mechanism. The current semantic matching models combine the advantages of various deep learning models to obtain richer semantic representation.

The development of semantic matching technology is largely restricted by the variety and quality of datasets. Currently, the most popular datasets are used to identify semantic relationships in retrieval and question answering. However, the semantic matching model can actually be extended to a wider range of text matching tasks. For example, the classification of patent citation relationship in this paper is one of the applications of semantic matching model, except that semantic relationship is not the entailment or QA, but whether one of the granted patents is cited by another pending patent.

Application of deep learning in patent field

Deep learning has been highly developed by researchers due to its excellent effects in computer vision, speech recognition, natural language processing and other research fields. Despite the increasing complexity and depth of the research on deep learning methods, the methods actually used in the application field are still simple. For example, deep learning methods used in patent text analysis are still confined to simple BP neural networks, other models such as CNN, RNN and their variants are rarely adopted for semantic mining. It is found that the patent research based on deep learning currently mainly includes three categories: patent clustering and classification, trend analysis of the technology development and the analysis of patent value. There are also some other studies including patent translation, infringement identification and patent similarity, which are very few compared with the three main categories.

Patent clustering and classification In order to better visualize the analysis results, Self-organizing mapping (SOM), an unsupervised learning method, was often used in the early research on the classification of patent categories. For example, Kohonen et al. (2000) constructed a self-organizing mapping network to map a large number of patents documents

on a two-dimensional map, and it showed the distribution of patents of different topics in the map to illustrate the effectiveness of the mapping results. In addition, Kohonen also explained the user interaction operation that could be realized after visualization with SOM. Lamirel et al. (2006) further proposed multi-viewpoint self-organizing mapping (Multi-SOM) neural network to extract information of different dimensions of patents, which enhanced the depth of patent information mining.

In recent years, researches have shown that supervised learning is generally superior to unsupervised learning in the classification method, and the results are controllable and easy to verify. Therefore, there are many researches on the classification of patent texts by means of supervised learning. According to the application of neural network in different positions on text classification, it can be divided into classifier and text coding representation.

(1) Neural network as classifier. In the study of using neural network as classifier, only a few classifier models such as CNN and RNN have been tried. For example, Kowsari et al. (2017) proposed a hierarchical text classification framework based on deep learning. Since the category of patent can be divided into different levels from coarse to fine, in order to solve this problem, he used three different classification models of RNN, CNN and deep learning to classification of different levels. Experiments show that the hierarchical classification method based on deep learning is generally better than the traditional method in the classification of 3 patent datasets. Most of the remaining studies still chose BP neural network, such as Li Shengzhen et al. (2010) used vector space model to express the title and abstract of patent, and input BP neural network to classify the IPC category of patent after screening features of chi-square statistics. Generally speaking, the research core and innovation of BP neural network as a patent classifier often lies in how to better deal with patent text features. Trappey et al. (2006) first used VSM to represent patent text, then analyze the correlation between text features, and predict the patent IPC category according to the occurrence frequency of related features and the correlation degree. Amy et al. Trappey et al. (2013) used the domain ontology model created by Protege to deduce the semantic concept probability of key phrases that often appear in domain-related patent documents. Then, the word frequency and concept probability of key phrases are used as input of the neural network.

(2) Neural network as text coder. The reason why research on neural network as text encoder is that deep learning can more intelligently learn and express the semantic relationship in text, while other machine learning methods may be adopted in the classification stage. Xia et al. (2016) used the sparse representation of patent text with VSM, and compressed the sparse feature vector through the sparse self-encoder. Then the deep trust network was used to further extract the deep features of the text, and Softmax classifier was input to determine the category of the patent text. Ma Shuanggang (2016) input Denoising Auto Encoder (DAE) to further extract the low-dimensionality representation of the patent text after the text representation and feature screening, and then used SVM to classify the patents. Hu Jie et al. (2018) input the word vector into CNN to represent the patent text, and then use the random forest algorithm to predict the patent category.

Technology development trend analysis Most of the research based on deep learning to analyze the development trend of patented technology is to predict the number of

patents. Some scholars believe that the number of patent applications is easily affected by external factors such as national policies and market competition, so the number of authorized patents can show the actual trend of technological development more stably. Ma Junjie et al. (2013) took the number of patent authorization of previous nodes as the model input. Then he adopted the BP neural network with the activation function as the prediction model, which better fits the development trend of the number of patent authorization. Different from other studies that take the number of patent authorization of previous time nodes as the model input, Ramadhan et al. (2018) took time itself as the input of artificial neural network to predict the number of patents in a certain field at this time, and drew the technology life cycle curve accordingly to reveal the development trend of technology with time. In order to make the input time nonlinearity, Ramadhan also designed and increased the dimension of the input vector to make the input of the other two dimensions become the second and third power of time.

If the time characteristic is added to the clustering and classification results of patent text, it can be used to express the development trend of specific technology in a certain field. Sung et al. (2017) constructed the Growing Cell Structures (GCS) network, mapping the patents to the two-dimensional topic map. Girvan Newman algorithm is used to decompose the mapping results into an appropriate number of topic categories and establish a snapshot of the topic category map at a certain interval, visualization is applied to observe the generation and development trend of a patented technology over time.

Patent value analysis Patent value assessment is one of the research emphases of patent intelligence analysis. It is of great practical significance to clarify patent value, such as screening core patents in a certain field, studying competitive intelligence, technology development strategy of enterprises and so on. The technical value of patent is related to some objective evaluation indexes in patent documents and subjective evaluation indexes defined by experts. Jiaojiao and Yun (2017) first extracted 10 high-value patents from massive patent texts by using 9 objective evaluation indexes including number of patent rights and number of invention and principal component analysis method. Then Delphi method was used to analyze these 10 patents to determine 11 subjective evaluation indexes from three aspects of technical characteristics, technical feasibility and technical utility and make quantitative evaluation. At the same time, the influence of experts' experience and familiarity on the evaluation results is excluded. Finally, entropy weight method is used to determine the weight of each index. Lee et al. (2018) designed 18 indicators from five aspects of novelty, scientific intensity, growth rate, scope of application and development ability, and predicted the number of cited patents in 3, 5 and 10 years through artificial neural network, thus reflecting the potential influence of patents.

Although the development speed of deep learning in application is not as fast as that of method research, the development of method can often drive the change of application field. In recent years, the research of deep mining text semantic information with complex neural network has gradually emerged in the field of patent. Xiaokang (2017) studied the difficulties in translation caused by out-of-set words in corpus that are not in the scope of dictionary. He used the Encoder–Decoder model in deep learning and added the mechanism of Attention alignment to assist the translation of out-of-set words, which achieved a good translation effect. On the basis of the definition of patent similarity in Patent Model Tree (PMT), Yuxiang (2014) calculated the patent similarity by using patent claim text and Siamese LSTM Model. The final similarity was the weighted sum of similarity of all PMT

nodes. However, the disadvantage of this study is that the sample size is small and the corpus used is the existing Quora data set rather than the corpus used in this study.

From the above discussion, it can be found that since 2018, there have been relevant studies on patent semantic similarity and semantic matching at home and abroad. The research on classification and similarity of patent citation relation in this paper is an extension of the above research.

Methodology

This part consists of methodology on patent citation classification and patent technology similarity based on deep learning. First, we proposed the model architecture of patent citation classification and elaborated on specific methods of the model. Then, we proposed the axiom and hypothesis on patent technology similarity. According to that, we used the classification probability of patent citation to represent the technology similarity of two patents. Besides, we illustrated the rationality of the hypothesis by statistics method and manual spot check.

Patent citation classification model based on deep learning

Figure 1 is the basic model architecture diagram of patent citation classification. The specific methods of each part in the model are described in detail below.

Context word embedding generated by bidirectional RNN coding Figure 2 is the RNN text encoding process of this model, where the input vector $X^{[t]}$ represents the word embedded at time t or position t of a document, which is trained by word2vec. The purpose of selecting bidirectional RNN to encode the patent document is to obtain the embedded representation of words in the document containing the context information, i.e., the hidden state $h^{[t]}$.

RNN has two common variants: Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU). LSTM was proposed by Hochreiter and Jrgen (1997) in 1997. At that time, the design of LSTM unit did not include forget gate, and it was not until Alex Graves (2008) improved the LSTM at that time and added the design of forget

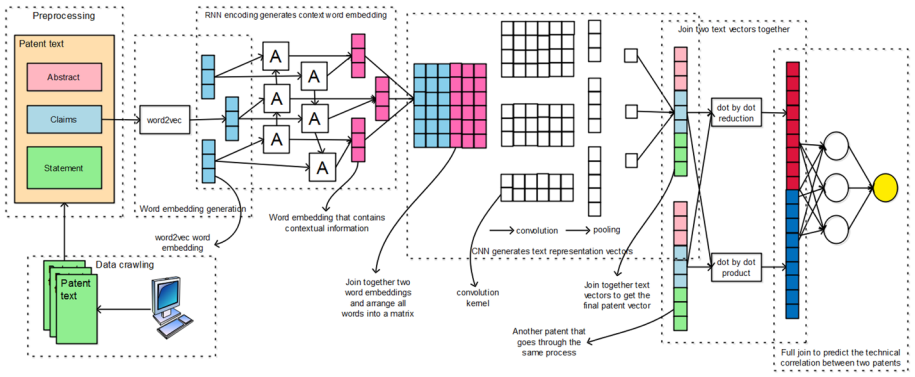

Fig. 1 The basic model architecture of the classification of patent citation relation based on deep learning

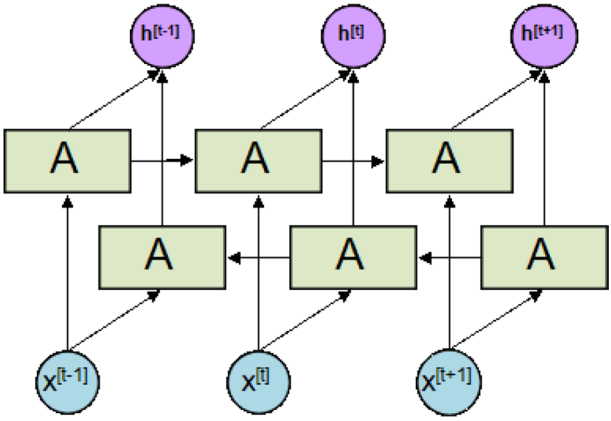

Fig. 2 RNN text encoding process

gate that the most common LSTM was formed. Compared with RNN, the advantage of LSTM is that it can avoid the long-term dependency problem through deliberate design, and LSTM can use the gate function to solve the vanishing gradient problem. GRU was proposed by Cho et al. (2014) in 2014. Its basic unit design originates from LSTM, but it is simpler and easier to implement than LSTM, and it can achieve the same effect as LSTM in the experiment. In order to solve the gradient descent problem and at the same time ensure the operation efficiency of the algorithm, this model selects bidirectional GRU to embed the input words into the context. The specific process is as follows.

The input at time t is composed of the input $X^{[t]}$ of the sample sequence at time t and the hidden state $h^{[t-1]}$ at time $t-1$. The input vector first enters a sigmoid hiding layer, then outputs a gate vector $h_r^{[t]}$ for filtering information. This door is called the reset door, which is used to filter the hidden state at time $t-1$ and determine the part used to update the hidden state at time $t-1$. The process is expressed by the formula as shown in Eq. (1) :

$$h_r^{[t]} = \sigma\left(w_r \cdot \begin{bmatrix} X^{[t]} \\ h^{[t-1]} \end{bmatrix} + b_r\right) \tag{1}$$

where $h_r^{[t]}$ is the reset gate vector at time t; $X^{[t]}$ is the input vector to the sample sequence at time t; w_r is the weight matrix between the input layer and the reset gate sigmoid hidden layer, which is shared in each GRU unit; the sigmoid function represents the calculation of the sigmoid function value for each element in the vector of independent variables.

At the same time, $X^{[t]}$ and $h^{[t-1]}$ as input vectors enter into another sigmoid hidden layer, then a gate vector $h_{fu}^{[t]}$ for filtering information is the output. This gate is called the update gate and acts as both the update and forget gate in the LSTM unit. On the one hand, the information retained in the hidden state at time $t-1$ is screened out; on the other hand, the input information used to update the hidden state at this time is screened out. The process is expressed by the formula as shown in Eq. (2):

$$h_{fu}^{[t]} = \sigma\left(w_{fu} \cdot \begin{bmatrix} X^{[t]} \\ h^{[t-1]} \end{bmatrix} + b_{fu}\right) \tag{2}$$

where $h_{fu}^{[t]}$ is the update gate vector at time t; $X^{[t]}$ is the input vector to the sample sequence at time t; b_{fu} is the offset vector of update gate sigmoid hidden layer, which is shared in each

GRU unit; the sigmoid function represents the calculation of the sigmoid function value for each element in the independent variable vector, i.e. point by point sigmoid calculation.

The hidden state at time $t-1$ filtered by reset gate and the input of sample sequence at t moment jointly constitute the input information vector. The vector enters a tanh hiding layer, which encodes the input information vector and finally gets the input information for updating the hidden state. The process is expressed by the formula as shown in Eq. (3) :

$$h_i^{[t]} = \tanh\left(w_i \cdot \begin{bmatrix} X^{[t]} \\ h_r^{[t]} \cdot h^{[t-1]} \end{bmatrix} + b_i \right) \tag{3}$$

where $h_i^{[t]}$ is the input information at time t; $h_r^{[t]}$ is the reset gate vector at time t; $X^{[t]}$ is the input vector to the sample sequence at time t; w_i is the weight matrix between the input layer and tanh hidden layer, which is Shared in each GRU unit; b_i is the offset vector of tanh hidden layer, shared in each GRU unit; tanh function refers to the calculation of tanh function value for each element in the vector of independent variables, i.e. point-by-point tanh calculation.

Next, on the one hand, the input information is filtered by the update gate to obtain the information for updating the hidden state. On the other hand, point by point 1-operation is carried out on the update gate to obtain another forget gate, and the hidden state of the previous moment is filtered to determine the reserved information. The sum of the two pieces of information is the hidden state updated at time t. In fact, the function of update gate is equivalent to assigning a weight to the input information and the hidden state at the previous moment, and the weight sum is 1. The process is expressed by the formula as shown in Eq. (4):

$$h^{[t]} = \left(1 \ominus h_{fu}^{[t]}\right) \otimes h^{[t-1]} \oplus h_{fu}^{[t]} \otimes h_i^{[t]} \tag{4}$$

where $h^{[t]}$ represents the hidden state at time t; $h_{fu}^{[t]}$ is the update gate vector at time t; $h_i^{[t]}$ is the input information at time t. $\ominus$ means point by point minus; $\otimes$ means point by point multiply; $\oplus$ means point by point add, which is the same as adding vectors.

This is a bidirectional GRU coding model, We express the hidden state output at the time of forward GRU encoding t as $\overrightarrow{h}^{[t]}$, and the input at the time of inverse GRU encoding t as $\overleftarrow{h}^{[t]}$. The word embedding at time t of the text is ultimately represented as the combination of the input vector $X^{[t]}$ of the sample sequence at time t and the context coding $\overrightarrow{h}^{[t]}$ and $\overleftarrow{h}^{[t]}$, as shown in Eq. (5) :

$$w^{[t]} = \begin{bmatrix} \overrightarrow{h}^{[t]} \\ X^{[t]} \\ \overleftarrow{h}^{[t]} \end{bmatrix}^T \tag{5}$$

Document vector representation generated by CNN The Fig. 3 is the CNN text encoding process of this model. Where $w^{[t]}$ is the word embedded final representation at the time t in Eq. (5), the dimension of it is K, $d_1^{'}$ is the initial representation of a document labeled 1, which is concluded by words embedded of the document arranged in chronological order; the dimension of it is $n \times k$, n is the number of words in the document. h_1, h_2, h_3 is the height of the three kinds of convolution kernels; k is the width and m is the number of each type of convolution kernel. The reason we choose different height of convolution kernels

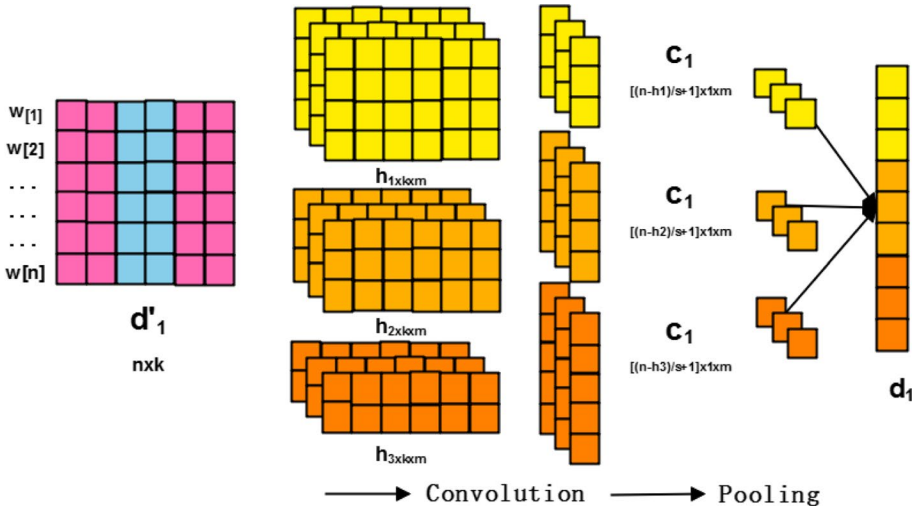

Fig. 3 CNN text coding process

is to consider the semantic relationships between text of different lengths in a document. Each height has multiple convolution kernels in order to extract semantic features from different aspects of the text. c_1 represents the output result of convolution kernel with height h_1, which is composed of $m\,(n-h_1)/s+1$ dimension vectors. d_1 is the final text representation result, and the dimension is $3 \times m$. The specific calculation process of the model is as follows:

First, the initial representation of a document is shown in Eq. (6):

$$d_i' = \begin{bmatrix} w_i^{[1]} \\ w_i^{[2]} \\ \cdots \\ \cdots \\ w_i^{[n]} \end{bmatrix} \tag{6}$$

Then d_i' needs to extract features through the convolution layer. In CNN convolution layer, there are three convolution kernels of different heights, and the number of each convolution kernel is m. The convolution and pooling process of CNN model is explained below with the first convolution kernel of the first kind of height as an example. The characteristic diagram generated in the convolution process is shown in Eq. (7):

$$c_{11} = \left[c_{11}^{[1]} \; c_{11}^{[2]} \; \cdots \; \cdots \; c_{11}^{[(n-h_1)/s+1]} \right] \tag{7}$$

where c_{11} represents the characteristic graph generated by the first convolution kernel at the first height; $c_{11}^{[t]}$ denotes the eigenvalue of the t position of the convolution kernel; $(n-h_1)/s+1$ show the dimensions of the feature graph, where n is the number of words in the document; h_1 is the height of the convolution kernel; s is the step size of the convolution kernel. The calculation method of $c_{11}^{[t]}$ see Eq. (8):

$$c_{11}^{[t]} = \mathrm{ReLU}\left(\mathrm{flat}(w_{11}) \cdot \mathrm{flat}\left(\begin{bmatrix} w_i^{[t]} \\ w_i^{[t+1]} \\ \cdots \\ \cdots \\ w_i^{[t+h-1]} \end{bmatrix}^T\right) + b_{11}\right) \tag{8}$$

where $\mathrm{flat}(\omega_{11})$ denotes the result of flattening the weight matrix of the convolution kernel into a vector; b_{11} represents the offset part of the ReLU activation function. The eigenvalue of t position is obtained through the inner product of the flattened vector of the convolution kernel weight matrix, the partially flattened vector of the t to $t + h - 1$ row of the document matrix $d_i^{'}$, bias and ReLU activation unit processing.

Then the feature graph c_{11} is maximally pooled to represent the most important feature in the convolution kernel. As shown in Eq. (9) :

$$\hat{c}_{11} = \max\left(c_{11}\right) \tag{9}$$

The feature graph generated by m convolution kernels of the first kind of height is finally pooled to generate m features, like Eq. (10):

$$\hat{c}_1 = \left[\hat{c}_{11}, \hat{c}_{12}, \ldots \ldots \hat{c}_{1m}\right]^T \tag{10}$$

The final text representation results are obtained by splicing the eigenvector results of the three convolution kernels, as shown in Eq. (11):

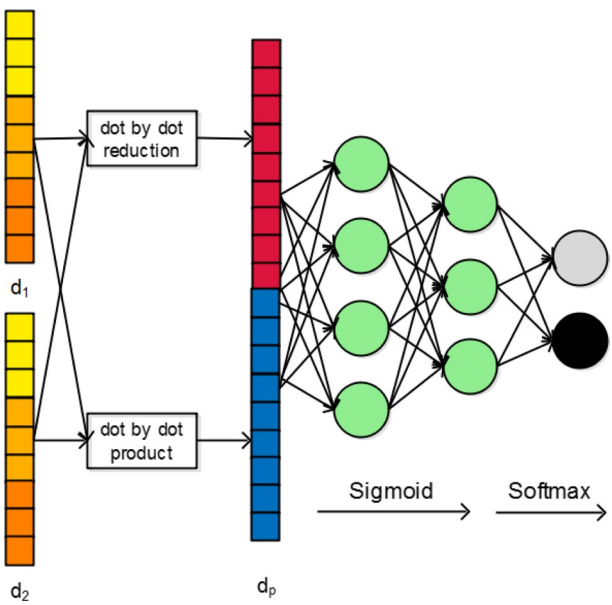

Fig. 4 MLP coding and Softmax classification

$$d_i = \begin{bmatrix} \hat{c}_1 \\ \hat{c}_2 \\ \hat{c}_3 \end{bmatrix} \tag{11}$$

MLP coding and Softmax classification Multi-layer perceptron (MLP) is also known as Artificial Neural Network (ANN) and Back Propagation (BP) Neural Network. It is composed of three parts: input layer, hidden layer and output layer. Neurons in each layer are connected with neurons in adjacent layers in pairs. Therefore, it can also be called a fully connected neural network. Figure 4 is the matching part of this model, which is composed of text pair vector splicing, MLP neural network coding and Softmax classifier.

The ultimate goal of this model is to determine whether there is a citation relationship between two documents. Therefore, it is not enough to encode the patent documents, and it needs to integrate the two patent documents to be tested together for citation classification. The citation relation referred to in this paper is undirected, i.e., if one of the two patents cites the other, it is deemed that there is a citation relation between them, and it is not necessary to identify which one is cited or citing. Based on this premise, it is inappropriate to combine the representation vectors of two documents directly, because the sequence of the document vectors may affect the output. Referring to the heuristic matching method of Mou et al. (2015), this model obtains the integrated representation of two documents through the operations of point-by-point multiplication and point-by-point subtraction of the representation vectors of the two documents, and because the relationship is undirected, the vectors after point-by-point subtraction also undergo point-by-point absolute value processing. As shown in Eq. (12) :

$$d_p = \begin{bmatrix} \left| d_i \ominus d_j \right| \\ d_i \otimes d_j \end{bmatrix} \tag{12}$$

Among them,d_p is the integration of two pending patent documents. d_i and d_j are vector representations of two patents encoded by CNN. $\ominus$ denotes point-by-point subtraction of vectors; $\otimes$ denotes point-by-point multiplication of vectors.

Then d_p, the integrated document pairs are input into the artificial neural network containing several hidden layers for feature coding. The number of hidden layer and neurons in each layer will be determined during the experiment, but the basic principle is to decrease the gradient layer by layer to avoid the mutation of the number of neurons between adjacent layers. This is done to compress the feature information of the document pair step by step, and to avoid the sudden loss of too much semantic information between adjacent layers. If there are i hidden layers in the artificial neural network, the output of the first layer is shown in Eq. (13):

$$y_1 = \mathrm{ReLU}\left(\omega_1 \cdot d_p + b_1\right) \tag{13}$$

y_1 is the output vector of the first layer; ω_1 is the weight vector of the first hidden layer. b_1 is the bias vector of the first hidden layer; the activation function is ReLU function. Similarly, the output of layer i is shown in Eq. (14):

$$y_i = \mathrm{ReLU}\left(\omega_i \cdot y_{i-1} + b_i\right) \tag{14}$$

The output of layer i also needs to enter the last Softmax classification layer. Since this research involves binary classification, the Softmax classification layer only has two neurons, as shown in Eqs. (15) and (16):

$$\hat{y} = \text{Softmax}(z) \tag{15}$$

$$z = \omega \cdot y_i + b \tag{16}$$

The probability to predict the citation relation between two patents is shown in Eq. (17):

$$\hat{y}_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_0}} \tag{17}$$

Among them, $\hat{y}_1$ is the probability of predicting the citation relation between two patents; z_1 and z_0 are two elements of vector Z in Eq. (16), which are respectively used to predict the probability of whether there's a citation relation between two patents. In order to optimize the proposed model during the training process, it's very necessary to reasonably construct the loss function of the model. In this paper, the most common cross entropy loss function is selected, as shown in Eq. (18):

$$\text{Loss} = -\sum_{i=0,N} y_i \log \hat{y}_i \tag{18}$$

since $y_0 = 0$, the loss function is finally shown in Eq. (19):

$$\text{Loss} = -\log \hat{y}_1 \tag{19}$$

Patent technology similarity representation

Axioms and hypotheses Chinese patent law stipulates the granted patent should be novel, creative and practical, which means that the patent cannot be an existing technical scheme when applied. It should have some improvements compared with other relevant technologies at that time, and it can be put into use and produce beneficial effects. According to part 1, we know that patent citations come not only from applicants but also mainly from patent examiners. Applicants tend to not cite or refute other patents with similar technologies because they may damage novelty, creativity and practicability. In order to examine an applied patent, examiners must provide "comparison document" as reference. These "comparison document" should be other patents whose technology is similar to the applied patent, and serve as the examiners' supplementary citations. So the axiom follows from the above discussion:

Axiom: If there is an examiner citation relation between two patents, the two patents must be similar in technology.

The classification model of patent citation can determine whether two patents exist examiner citation relation, which can judge whether the two patents are similar in technology by the axiom. The output of Softmax classifier is probability of whether there's examiner citation relation. So the probability of examiner citation relation (i.e. the probability of technology similarity) may be able to measure the technology similarity of two patents. Tai et al. (2015) and Zhou et al. (2016) used the deep learning model to divide the patent technology correlation into 5 categories, and the correlation score is the weighted sum of the score represented by these 5 categories and the probability of the corresponding category. Similarly, in this

paper, the category with examiner citation relation (i.e. technology similarity) is used to represent the score 1, while the category without examiner citation relation (i.e. technology dissimilarity) is used to represent the score 0. The weighted sum of the score value represented by these two categories and the probability of the corresponding category is the probability with examiner citation relation (i.e. technology similarity). Based on the above discussion, the following assumptions are proposed.

Hypothesis: *The classification probability of patent citation based on deep learning can represent the technology similarity (examiner citation relation) of two patents.*

In order to verify this hypothesis, the following part will research the validity of the classification probability of patent citation as patent technology similarity by the means of machine statistics and manual sampling analysis.

Reasoning and validation (1) Machine statistical method. If there are 3 patent documents named A, B, C where A exist citation relation with B but doesn't exist with C, the technology similarity of A and B should be higher than that of A and C. If the citation relation probability of A and B is higher than that of A and C, it indicates that the classification probability of patent citation can represent the technology similarity of two patents.

We construct a large number of patent triples mentioned above, and calculate the probability of citation relation between patents through the classification model of patent citation relation. If the probability of citation relation between A and B is higher than that between A and C, we believe that this probability is correctly judged once as the technology similarity of the patent. Based on this, we calculate the accuracy of this probability of judging technology similarity in all triples and compare it with other methods.

(2) Manual sampling analysis. The patent citation classification model produces four classification situations, respectively, there exist citation relation and the prediction is correct (TP), there exist citation relation but the prediction is wrong (FN), there not exist citation relation and the prediction is correct (TN), there not exist citation relation but the prediction is wrong (FP). We select some patent pairs randomly from the four classification results, and analyze their technology similarity manually. If the patents in TP category were more technically similar than those in TN, it indicates that the patent pairs with higher probability of citation relation were more technically similar than those with lower probability. However, the technology similarity of patent pairs in FN and FP categories needs to be analyzed according to actual situation. If patent pairs in FN are more technically similar than those in FP, it indicates that the misclassification of citation relation affects the value of the technology similarity. If patents pairs in FP are more technically similar than those in FN, the annotation of the original data set may be inaccurate and some patent pairs with technology similarity are not referenced by the examiner.

Besides, all patents referenced in a certain patent in the test set are extracted. We respectively used the proposed method in this paper and the cosine method to calculate the similarity between referenced patents and the selected patent, and ranked the referenced patents according to the two results. We analyze the two disparate patents ranked by two methods. Assuming that the similarity of patent A calculated by the proposed method is higher than that of patent B, while the cosine method does the opposite. If the technology similarity between patent A and the selected patent is higher than patent B through manual analysis, then the proposed method can effectively measure the technology similarity of the patent.

Experiments

Data acquisition and preprocessing

The selenium package in python with Firefox browser is used to crawl the full text of patents in a specific field on Google Patent. Finally, two patent full text data sets were crawled for patent citation relation classification and similarity analysis. The first patent dataset obtained by using "machine learning" and "text" as keywords belongs to the field of text processing. The second patent dataset obtained by setting the International Patent Classification (IPC) number as H04 in advanced retrieval is the field of telecommunications technology. The crawl content contains 12 fields, as shown in Table 1.

The most important step of data preprocessing is segmentation. Before word segmentation, we filter the collected data to remove the patent text with field missing caused by the problem of network element recognition. In order to achieve better segmentation effect, we first use the python toolkit pynlpir of Chinese academy of sciences word segmentation software (NLPIR) to extract the keywords in the patent text, and then add user dictionary to assist word segmentation process after manual screening. In the word segmentation process, we select jieba segmentation system that is simple to operate and efficient to segment. After that, we import stop list to filter stop words in the text, as well as punctuation marks, numbers and other non-Chinese characters.

In order to carry out the subsequent experiments smoothly, it is necessary to generate patent pair samples for training and testing models in the data preprocessing, including patent pairs with and without citation relation. The way to generate patent pair with citation relation is as follows: iterating through all patents in the dataset, obtaining the citing and cited patent list of each patent A, pairing each patent B in the list with patent A, and taking

Table 1 Patent full text data crawling field and its description

Field name	Field description
Patent_no	Patent number, one of the unique identifiers of the patent, is composed of application country code, application type and the serial number
Title	Title of patent
Abstract	Abstract of patent
Technical_field	The technical field to which a patent belongs
Background	Background technology, including the statements and shortcomings of the existing technology
Invent_content	Description of the invention, including the unresolved technical problems and the proposed technical scheme
Benefit	The beneficial effects compared with existing technologies
Invent_mode	The specific implementation mode which describes all details of technical implementation and lists one or more preferred embodiments
Claims	The right to apply for a patent to be protected according to the statement of contents, which have a strict writing format and standard terminology
Citations	The list of patents cited by this patent, especially referring to examiner citation
Cited	The list of patents that cite this patent, especially referring to examiner citation
Similar	The list of similar patents which is identified by Google, but excludes patents that have a citation relationship with this patent

Table 2 Basic statistics of the dataset after preprocessing

Topic	Dataset1: text processing	Dataset2: IPC H04
Number of patents	17,274	38,331
Patent pairs with citation relation	46,315	74,308
Patent pairs without citation relation	28,537	137,498
Training set: test set	9:1	9:1

the non-repeated patent pairs as the sample with citation relation. The way to generate patent pair without citation relation is as follows: iterating through all patents in the dataset, obtaining the similar patent list of each patent A, pairing each patent B in the list with patent A, and taking the non-repeated patent pairs as the sample without citation relation.

Table 2 shows the basic statistics of the dataset after preprocessing, including the number of valid patents, patent pairs with/without citation relation and the quantity ratio of training set versus test set.

Patent citation classification

Model input By analyzing the crawled patent texts, the following conclusion can be found: (1) among all text fields of the patent, which can best reflect the technical content is the invention content, specific implementation mode and claim. (2) Much of the invention content, specific implementation mode and claim contents is repeated. (3) Among them, the specific implementation mode adds more technical implementation details than the invention content, but the differences between implementation details and examples don't affect whether the patent will be cited. (4) The claim is organized and written based on the specification, which clearly and briefly describes the technical characteristics of the patent in a standardized format. (5) The examiners cite patents on the basis of whether the right to apply for a patent infringes upon the rights of some prior patent. From the above conclusions, it can be seen that the patent claim has the greatest impact on whether the patent is cited by the examiner, and the claim is more standardized and concise than the invention content and specific implementation mode. Therefore, in order to reduce the model training time, this paper only selects *patent claim* as the input part to be more accurate and efficient.

Model combination The classification model of patent citation based on deep learning uses Word2vec to obtain word general embeddings at the lexical coding level, which proceed to CNN document encoding process together with the word embeddings containing context information obtained through Bidirectional GRU encoding. Finally, we use MLP neural network to further extract and compress the features, and obtain the final classification results through Softmax. The Bidirectional GRU is set at the lexical coding level to consider the influence of the order of contextual words on the expression of words, because Word2vec only considers the words in the context fixed window in the training process, but not the word order. The reason why CNN is selected in the document coding level is that CNN can not only effectively extract and compress the document global features, but also further consider the semantic relation between sentences or paragraphs by designing the shape and moving step of the convolution kernel. In the model interaction and classification part, we introduced the MLP neural network to gradually compress and extract

the features most relevant to the classification task due to its ability of fitting any nonlinear relationship.

However, in the actual experiment, the above model combination may not achieve the best classification effect. For example, both the Bidirectional GRU and the CNN coding process take the context information into account, then the overlapping of these functions may lead to the waste of model running resources. And if the CNN coding process has been able to obtain a better document representation, then using MLP to further extract features is a redundant step. In order to obtain the best model combination, this paper will examine the influence of each part of the model on the experimental results by deleting Bidirectional GRU and MLP neural network. But the CNN coding process is an indispensable step to compress the document matrix into a vector, so it will always be retained in the experimental model. If the classification effect of the model increases after the removal of Bidirectional GRU or MLP parts, it indicates that these parts are unnecessary for the model construction.

Hyper-parameter setting All parameters of our model (such as weight vector) are acquired automatically all in training process, but the hyper-parameters need to be set up manually according to experience and actual situation before training Kim (2014). Table 3 lists the main types of hyper-parameter involved in the model and their initialization.

Comparative experimental model

In order to compare the proposed model with traditional text representation and classification methods, this paper sets up two sets of comparative experimental models. One is the models composed of TF–IDF document representation and various machine learning classification methods which is also the baseline of this paper, the other is the models composed of Doc2vec document representation and various text classification methods.

TF–IDF document representation TF–IDF representation of patent text is completed by feature extraction module of sklearn library. Because TF–IDF representation of the large

Table 3 The main types of hyper-parameter involved in the model, their descriptions and initialization

Hyper-parameter	Description	Initialization
Size	The dimension of word embedding generated by Word2vec	50
Window	The maximum distance between the Word2vec current word and the context word considered by model	8
rnn_hidden	The dimension of the hidden vector of RNN	25
Filters	The number of CNN convolution kernel	50
kernel_size	The shape of CNN convolution kernel	[40, 60, 80]
Strides	The step size of CNN convolution kernel	20
hidden_layers	The number of the hidden layers of MLP	3
mlp_hidden	The number of neural units in the hidden layers of MLP	[150,100,50]
N	Document length, that is, the maximum number of words contained in the document	5000
learning_rate	Model learning rate, used to optimize the model training process	0.0005
batch_size	The number of samples included in each batch of training data	32
Epoch	The number of iterations of training data	20

size of texts will consume a lot of running resources, so the program will select the 10,000 words with the highest word frequency. The 10000 text features also need to be further screened to obtain what features contribute most to the classification task. The text representation of the proposed model encoded by CNN has 150 dimensions. But it is obviously not sufficient to select only 150 out of the 10,000 features represented by TF–IDF, so the paper will adopt the CHI square feature selection method to select 1000 features for text classification.

Doc2vec document representation The Doc2vec representation of patent text is completed by Doc2vec module of gensim library, where the hyper-parameters in the training process mainly include the selected training method, the context window and the dimension of the generated document representation. We select PV–DM method to train the Doc2vec embedding of patent, and set the context window to 10 and the dimension of generated document representation to 150 the same as that encoded by CNN.

Supervised learning methods Before the classification of patent citation, the document representation of patent text pair to be tested is also concatenated according to formula (12). The concatenated representation is combined with the following four common machine learning text classification methods: (1) K-Nearest Neighbor (KNN), that is to judge its category according to that of the most recent K samples of the current sample. (2) Naive Bayes (NB), that is a probabilistic generation model based on Bayes theorem and independent assumptions of characteristic conditions. (3) Support Vector Machine (SVM), that is to divide into categories by constructing the hyperplane furthest away from the sample. (4) Multi-Layer perceptron (MLP), that is a model which produces good fitting results for any nonlinear problem. The above four classification methods are also implemented by programming the relevant modules of sklearn library. Among them, KNN, NB and SVM select 10-fold cross-validation method to obtain the most stable evaluation results.

Results

Model assessment

In this paper, the error rate that is the simplest and can reflect the overall classification effect of the model is selected as the evaluation index. As shown in Table 4, the error rate

Table 4 Error rate of classification model and comparison model based on deep learning (%)

Model	Dataset 1	Dataset 2
TF–IDF+KNN	21.343443	24.632617
TF–IDF+NB	31.451392	38.126397
TF–IDF+SVM	38.085824	35.083047
TF–IDF+MLP	21.440021	26.830650
Doc2vec+KNN	24.625929	23.708016
Doc2vec+NB	42.243349	39.848726
Doc2vec+SVM	29.591729	7.733964
Doc2vec+MLP	39.153086	32.444172
GRU+CNN+MLP	11.180871	10.282801
CNN+MLP	**9.070264**	**9.451890**
CNN	10.619824	12.907825

of the proposed model and its comparison model on the two data sets is presented. It can be found that the combination model of CNN and MLP has the lowest error rate and performs best, which is the proposed classification model in this paper. Dataset 1 belongs to the field of text processing and Dataset 2 belongs to the field of telecommunications technology of IPC H04.

Firstly, the deep learning model of bidirectional GRU, CNN and MLP is trained and evaluated. In order to find the influence of each part of the model on the final classification result and obtain the optimal model combination, this paper removes one part of the model while keeping the other parts unchanged, and then evaluates the new model after training. The experimental results show that after the removal of bidirectional GRU part, the classification error rate of the model on both data sets is slightly reduced. Therefore, bidirectional GRU coding on the classification model should be removed from the model combination. In order to explore whether the MLP part has played a positive role in the model combination of CNN and MLP, this paper further removed the MLP part of the model. After testing on the two data sets, it was found that the classification error rate of the model increased after MLP was removed, which indicated that the MLP model was a necessary part of the patent citation classification model based on deep learning. Therefore, the best combination of models is to use CNN to encode documents and enter MLP to complete the classification of citation relationship.

In addition, Table 4 shows a comparison model of classification error rate is between 20% ~ 40%, higher than this proposed model by 10–30%. It also indicates that the classification effect of the model in this paper is generally better than the traditional method. However, the combination of Doc2vec document representation and SVM classification method has achieved better experimental results than the model in this paper on data sets in the field of electrical communication technology, and the error rate is about 1.7% lower than the model in this paper. This paper believes that this conclusion cannot represent that the model combination of Doc2vec and SVM is more effective than the model classification proposed in this paper. On the contrary, it indicates that the classification results of the model in this paper are more stable on the two data sets than other traditional methods. Because the error rate of this model in the field of text processing data set is nearly 20% higher than that of the model in this paper, and Doc2vec representation method and SVM classification method have not produced particularly good results in other experiments, which indicates that the excellent performance of this model in the field of electrical communication data set may be a special case.

According to the experimental results, the following conclusions can be drawn in this paper: (1) compared with the traditional text representation and classification methods, the deep learning-based patent citation relation classification model proposed in this paper can achieve lower model errors. (2) Classification results of the model proposed in this paper are more stable on different data sets than other models.

Similarity analysis of patent technology

According to Sect. 3, this paper proposes the following hypothesis: the classification probability of patent citation obtained by the model in this paper on two data sets can represent the technology similarity of patent pairs in the data set. The rationality of this hypothesis will be explained through machine statistics and manual analysis.

Table 5 Accuracy of each similarity calculation method to judge the technology similarity of patent (%)

Model	Dataset 1	Dataset 2
TF–IDF+Cosin	68.672956	66.474842
Doc2vec+Cosin	61.559121	61.023120
CNN+MLP	**93.898492**	**87.461805**

Machine Statistical Method According to the method in Sect. 3, triples are used to verify the accuracy of the probabilistic judgment technology of patent citation relation. In order to show that the probability of patent citation relation proposed in this paper can better represent the patented technology similarity than that obtained by other similarity calculation methods, this paper also introduces cosine similarity calculation method to judge the technology similarity of triples. As shown in Table 5, the accuracy of the proposed method in this paper is more than 25% higher than that of the cosine method whether TF–IDF or word2vec is selected as the text representation method. It indicates the proposed model performs best in below similarity calculation.

Manual sampling analysis In order to reduce the workload of manual analysis, this paper only takes the dataset of text analysis field as an example to analyze the technology similarity of patent pairs.

(1) Analyze the results in the confusion matrix.

According to the classification results in Sect. 5, three patent pairs were randomly selected from data sets in the field of text analysis, including positive sample predicted to be positive (TP), negative sample predicted to be positive (FP), positive sample predicted to be negative (FN) and negative sample predicted to be negative (TN). Then we analyzed technology similarity of three patent pairs for each kind, as shown in Table 6.

Among them, in TP-type patent pairs, the technology proposed by citing paper may be a subset of the cited paper (CN101004737A*CN104077011A), or it is highly similar to the cited paper (CN101018137A*CN103294466A), or it adopts a similar solution in the intersection of the technical field (CN101013443A*CN102193639A). In TN-type patent pairs, they may be related to each other in the technical field to some extent (CN101017428A*CN102651217A and CN101021838A*CN104123291A), but the technical problems and solutions concerned have no similarities. In addition, from the perspective of numerical results, the technology similarity of the patent pair CN101017428A and CN102651217A in the same field of speech is lower than that of the other two patent pairs, possibly because CN101013443A and CN101079031A are also related to word generation, while CN101021838A and CN104123291A are related to the text processing method in the text classification process. This shows that the method proposed in this paper can avoid the impact of large technical fields on technology similarity, but pay more attention to the implementation details of methods and technologies.

The above is about technology similarity analysis of patent pairs with correct classification of patent citation relation. However, some patents are misclassified by the model. In FP-type patent pairs, there is a great similarity between the solved problems and proposed technical solutions by CN101090371A and CN102882930A, but CN102882930A did not cite CN101090371A during the examination, which may be related to the withdrawal of the former application shortly after substantive examination. The other two patent pairs are not similar in technical implementation, but both CN101102316A and CN101178714A are related to web text processing technology, which may be the cause of misjudgment. In FN-type patent pairs, in addition to the fact that CN101072168B and CN102780644A are

Table 6 Patent pairs extracted from various results and their technology similarity

TP		FP	
Patent pair	Technology similarity (classification probability)	Patent pair	Technology similarity (classification probability)
CN101013443A*CN102193639A	0.97274	CN101090371A*CN102882930A	0.999975
CN101004737A*CN104077011A	0.994347	CN101101605A*CN104899324A	0.998172
CN101018137A*CN103294466A	0.999999	CN101102316A*CN101178714A	0.999994
FN		TN	
Patent pair	Technology similarity (classification probability)	Patent pair	Technology similarity (classification probability)
CN101072168B*CN102780644A	1.40×10^{-5}	CN101013443A*CN101079031A	0.000844197
CN101087259A*CN102682037A	0.00670322	CN101017428A*CN102651217A	8.14×10^{-11}
CN101114295A*CN104239373A	0.16196	CN101021838A*CN104123291A	0.00113216

indeed similar in technology but misjudged by the model, the remaining two patents have no similarities in the technical field and implementation details, which may be inappropriate to cite in the substantive examination.

As shown above, the patent pairs with citation relationship that can be correctly identified by the model do have some similarities at technical solutions. And the patent pairs with no citation relationship that can be correctly identified may have similarities in the technical field, but not in technical implementation. However, the patent pairs wrongly identified by the model actually have citation relationship and technology similarity, but their original data set was not cited or improperly cited, which affected the classification effect of the model. It shows that the proposed technology similarity can clearly distinguish the patent pairs similar with each other or not. The higher similarity value is calculated for the patent pairs with similar technology and vice versa. However, whether this value can better quantify the degree of technology similarity needs to be further explored.

(2) Compare the results of technology similarity and cosine similarity sorting.

The reason why this paper use numerical value to measure the technology similarity of patent is for the consideration of practical application. The ultimate goal of the proposed model is to help examiners select comparable documents for similar technology from a large number of patents. In fact, whether the patents to cite or not requires further judgment by the reviewer through manual review. When the number of patents with similar technology is huge that will require lot of time to manually review. If the patents could be ranked according to level of similarity in technology, the reviewer will mainly pay attention to most relevant patents. If the output of the probability model in this paper can be proved better than cosine method of similarity measure of patent technology, then we don't need other method to rank the technology similarity patents, this will save the computing resources. Table 7 shows one patent's similarity with its citations.

From the table, CN104572770A was identified as the most similar patent to CN105138537A by the similarity method proposed in this paper. But it was considered as the least similar patent in Doc2vec&cosine similarity calculation method, and also ranked low in TF–IDF&cosine method. However, CN101655866A was identified as the second to last similar to CN105138537A by proposed method. But it was considered as the second and third similar to CN105138537A in the remaining two methods.

To further analyze the technical content proposed by the sleeted patent CN10513837A (hereinafter referred to as "A") and its citing patents CN104572770A (hereinafter referred to as "B") and CN101655866A (hereinafter referred to as "C"). Firstly, from the perspective of technical field, A and B belong to the topic discovery technology, while C belongs

Table 7 Technology similarity between CN105138537A and its cited patents

Patent number	Technology similarity (classification probability)	Ranking	Doc2vec+Cosin	Ranking	TF–IDF+Cosin	Ranking
CN104572770A	1	1	0.192911	6	0.083259	4
CN104199846A	1	2	0.378166	1	0.166877	1
CN101464898A	0.999999	3	0.196072	5	0.108761	2
CN102053978A	0.999996	4	0.306591	3	0.037685	6
CN101655866A	0.999969	5	0.334226	2	0.092499	3
CN103530316A	0.999914	6	0.241306	4	0.059423	5

to the term extraction technology. Secondly, from the perspective of application scope, both A and C are used in scientific literature, while B does not limit application scenarios. Thirdly, from the perspective of the main content, both A and C use a lot of words to describe the way to choice of keywords, while B focuses on the combination and separation of topics, and describes the subject words in a small space. Finally, from the perspective of technical implementation details, both A and B use the method of calculating the amount of information when selecting the topic words. While A and C consider the word frequency to select the topic words(or terms), but A extracts the low-frequency words as candidate words and selects keywords based on the amount of information, C always selects the repeated string of high frequency to extract candidate term as a result. Therefore, if you don't consider the semantic content, A and C seems to be more similar than A and B. But if you deeply understand the technical details of these three patents, you will find the technology similarity of A and B is higher than that of A and C.

As known from the analysis of the above, the citation relation probability of the output in this model can serve as the technology similarity of patents. It's better than the traditional cosine method to identify the similarities in technical details between different patents. Thus the technology similarity of patents can be more accurately measured.

Conclusion

To help the reviewer to use the automatically technical means retrieve the patents which has similar technology with the existing patents, and to assist judge patent novelty, inventiveness and practical applicability, also can improve review efficiency and reduce labor costs, so this paper proposes a patent citation classification model based on deep learning. This model take the data set as an example which is in the field of text analysis and telecommunication technology to analyze the effect of the model and rationality of using the classification probability output by the model to measure the similarity of the patented technology. As we can see from the experiment, the error rate of classification is lower than the traditional text representation and classification method by 10–30%, and the classification effect used in the date sets is more stable. In addition, the accuracy of the proposed method to judge technology similarity of more than 25% higher than the cosine method. The results of manual analysis shows that the proposed technology similarity method can clearly distinguish the patents similar or not, and also be able to identify their similar degree. The limitations and prospects of this paper are as follows.

Limitations

The parameter test is not detailed enough, and the model training is not sufficient Because the single experiment takes about 2–5 days, the optimal parameters cannot be obtained through multiple experiments. The optimal hyper-parameters can only be selected by experience and information obtained from monitoring of the training process. In addition, during the experiment, it occurred overfitting phenomenon. But when we use the dropout method to avoid overfitting, it will reduce classification effect, which may be caused by inadequate model training. However, considering that further training would cost more time, this paper chose to remove the dropout method to achieve a better classification effect.

Uneven distribution of similarity values This paper regards the output probability of proposed model as technology similarity of two patents. However, after investigating the

distribution of this value in the classification results, it is found that this value shows a trend of polarization. That is, the similarity value of most patent pairs is very close to 1 and 0, and there are few values distributed between 0 and 1. It shows that although the proposed method can significantly distinguish between similar and dissimilar patents and measure the degree of technology similarity, it is not obvious to distinguish the degree of technology similarity of the similar (or dissimilar) patent technology.

Inaccurate data annotation In this research process, no manpower was expended on data annotation. The required labels of patent citation relation was obtained from examiners citation relation actually existing in the patent. That is to say, the labeling work of this paper is completed by a large number of patent examiners in the patent substantive examination. This labeling method not only avoids a lot of human labor, but also ensures the professionalism of the labeling work. However, the analysis in Sect. 5 also found some problems, that is, the examiners citation was also missed or miscited. This indicates that the data set annotation used in this paper is not completely accurate, which may affect the effect of model training in this paper.

Prospects

Add patent pairs with no similarity relationship, results into three categories of technology similarity and technical domain similarity and dissimilarity. The research is a binary classification problem. In the sample, one patent pair has citation relation (technology similarity), and the other dosen't have citation relation (technical dissimilarity). However, these patent pairs without citation relationship are still recognized as similar patents by Google. In fact, there should be an inclusion relation between patent similarity and technology similarity. Technical domain similarity does not mean that the adopted technology is similar, but technology similarity means that the technical domain must be similar. Adding samples of patents with no similarity relationship can not only enhance the recognition ability of the model for patents, but also make the results of technology similarity calculation more accurate.

Transform the identification of patent citation relationship into that of citation relationship of specific items in the patent claim. The model proposed in this paper is to automatically identify technology similarity patents that may be cited with the application patent so as to improve the efficiency of examiners. But the most liberating way for examiners is to directly identify which provisions in the patent claim might be violated. In other words, the most beneficial improvement of this paper is to transform the identification of patent citation relationship into that of claims relationship.

Acknowledgements The authors warmly thank reviewers for their valuable suggestions. This research was supported by National Natural Science Foundation of China [Grant Number: 71373291]. This research was supported by the Science and Technology Planning Project of Guangdong Province (China) [Grant Number: 2016B030303003].

References

Cho, K., Van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., & Schwenk, H., et al. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. *Computer Science*, pp. 1724–1734.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint*arXiv:1705.02364.

Gilbert, G. N. (1977). Referencing as persuasion. *Social Studies of Science*, *7*(1), 113–122.
Graves, A. (2008). Supervised sequence labelling with recurrent neural networks. *Studies in Computational Intelligence*, p. 385.
Hochreiter, S., & Jrgen, Schmidhuber. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.
Huang, P. S. , He, X. , Gao, J. , Deng, L., & Heck, L. (2013). Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Conference on information & Knowledge Management, ACM*.
Jaffe, A. B., Trajtenberg, M., & Henderson, R. (1993). Geographic localization of knowledge spillovers as evidenced by patent citations. *Quarterly Journal of Economics.*, *108*(3), 577–598.
Jiaojiao, Z. H. A. N. G., & Yun, L. I. U. (2017). Research on technology foresight model based on Delphi method and BP neural network. *Science Technology and Industry*, *17*(12), 81–88. +94.
Jie, H. U., Shaobo, L. I., Liya, Y. U., & Guanci, Y. A. N. G. (2018). A patent classification model based on convolutional neural networks and rand forest. *Science Technology and Engineering*, *18*(06), 268–272.
Junjie, MA, Jianxin, YOU, Rui, LU.(2013). Prediction of the number of invention patent authorization in China based on improved wavelet neural network. *Science & Technology Progress and Policy*., (04).
Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751.
Kohonen, T., Kaski, S., Lagus, K., et al. (2000). Self organization of a massive document collection. *IEEE Transactions on Neural Networks*, *11*(3), 574.
Kowsari, K., Brown, D. E., Heidarysafa, M., Meimandi, K. J., & Barnes, L. E. (2017). HDLTex: hierarchical deep learning for text classification. In *IEEE International Conference on Machine Learning and Applications. IEEE*, pp. 364–371.
Lamirel, J. C., Shehabi, S. A., Hoffmann, M., & Francois, C. (2006). Intelligent patent analysis through the use of a neural network: Experiment of multi-viewpoint analysis with the multisom model. *Acl Workshop on Patent Corpus Processing*, *20*, 7–23.
Lee, C., Kwon, O., Kim, M., et al. (2018). Early identification of emerging technologies: A machine learning approach using multiple patent indicators. *Technological Forecasting and Social Change.*, *127*, 291–303.
Li, X. I. E., Yong, D. E. N. G., & Sumin, Z. (2012). A comparative study on paper and patent citation. *Journal of Intelligence*, *20*(04), 19–21.
Mou, L., Men, R., Li, G., Xu, Y., Zhang, L., Yan, R., et al. (2015). Natural language inference by tree-based convolution and heuristic matching. *Computer Science*, *2*, 130–136.
Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., & Ward, R. (2014). Semantic modelling with long-short-term memory for information retrieval. *arXiv preprint*arXiv:1412.6629.
Ramadhan, M. H., Malik, V. I., & Sjafrizal, T. (2018). Artificial neural network approach for technology life cycle construction on patent data. In *2018 5th International Conference on Industrial Engineering and Applications (ICIEA) IEEE*, pp. 499–503.
Rui, L. I., & Liansheng, M. E. N. G. (2009). On the problems in patent citation analysis. *Information studies: Theory & Application.*, *21*(7), 39–43.
Shen, Y., He, X., Gao, J., Deng, L., & Mesnil, Grgoire. (2014). A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, ACM*, pp. 101–110.
Shengzhen, L. I., Jianxin, W. A. N. G., Jiandong, Q. I., & Lijun, Z. H. U. (2010). Automated categorization of patent based on back-propagation network. *Computer Engineering and Design.*, *31*(23), 5075–5078.
Shuanggang, M. A. (2016). *The Study of Automatic Chinese Patent Classification Based on Deep Learning Theory and Method*. Jiangsu: Jiangsu University.
Sung, H. Y., Yeh, H. Y., Lin, J. K., & Chen, S. H. (2017). A visualization tool of patent topic evolution using a growing cell structure neural network. *Scientometrics*, *111*(3), 1267–1285.
Tai, K. S., Socher, R., & Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *Computer Science*, *5*(1), 36.
Trappey, A. J. C., Hsu, F. C., Trappey, C. V., & Lin, C. I. (2006). Development of a patent document classification and search platform using a back-propagation network. *Expert Systems with Applications*, *31*(4), 755–765.
Trappey, A. J. C., Trappey, C. V., Chiang, T. A., & Huang, Y. H. (2013). Ontology-based neural network for patent knowledge management in design collaboration. *International Journal of Production Research*, *51*(7), 1992–2005.
Xia, B., Baoan, L.I., Lv, X. (2016). Research on patent document classification based on deep learning. In *International Conference on Artificial Intelligence and Industrial Engineering*.

Xiaokang, Z. H. E. N. G. (2017). *Research on the Transalation of Out of Vocabulary Words in the Neural Machine Translation for Chinese and English Patent Corpus*. Beijing: Beijing Jiaotong University.

Yin, W., Schütze, H., Xiang, B., & Zhou, B. (2016). Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics*, *4*, 259–272.

Yuxiang, M. A. (2014). *Research on Intelligent Patent Infringement Retrieval Based on Neural Network*. Chaoyang: Beijing University of Technology.

Zhang, K., Chen, E., Liu, Q., Liu, C., & Lv, G. (2017). A context-enriched neural network method for recognizing lexical entailment. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Zhou, Y., Liu, C., & Pan, Y. (2016). Modelling sentence pairs with tree-structured attentive encoder. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 2912–2922.

Zhou, Y., Liu, C., Pan, Y. (2016). Modelling sentence pairs with tree-structured attentive encoder. *arXiv preprint*arXiv:1610.02806.