# Detecting Technological Recombination using Semantic Analysis and Dynamic Network Analysis

Lu Huang
School of Management and Economics
Beijing Institute of Technology
Beijing, China
huanglu628@163.com

Xiaoli Cao*
School of Management and Economics
Beijing Institute of Technology
Beijing, China
cxl163990307@163.com

Yijie Cai
School of Management and Economics
Beijing Institute of Technology
Beijing, China
m15001185835@163.com

Hang Ren
School of Management and Economics
Beijing Institute of Technology
Beijing, China
renhang0988@163.com

Tianbin Xing
School of Management and Economics
Beijing Institute of Technology
Beijing, China
erushiya@163.com

Peifeng Ye
School of Management and Economics
Beijing Institute of Technology
Beijing, China
1404015686@qq.com

## ABSTRACT

Recombinative innovation is the innovation generated by the combinations of existing technical elements or new technological characteristics. In this paper, we proposed a novel method for detecting technological recombination, which combines semantic analysis and dynamic network analysis. Firstly, the dynamic word embedding model is applied to generate the dynamic word vectors, and construct the dynamic keyword network. Then, the dynamic network link prediction method is trained to predict the future network and generate the technological recombination opportunity score, which represents the possibility of potential recombination between technologies. Finally, SLM community detection is combined with the PageRank algorithm to identify core keywords in communities of the future network, and then detect potential technological recombination candidates corresponding to core keywords. A case study on artificial intelligence domain demonstrates the reliability of the methodology, and the results provide guidance for enterprise managers and technical policymakers.

## KEYWORDS

Technological recombination · Dynamic word embedding · Link prediction · Semantic analysis · Dynamic network analysis

## 1 Introduction

Recombinative innovation is the innovation generated by the combinations of existing technical elements or new technological characteristics [1], which has been considered a crucial way of innovation [2]. For example, OpenAI, an American artificial intelligence company, recombined natural language processing technology and computer vision technology to propose the DALL-E model, which is a multimodal transformer language model and improved the generation capabilities from text to image [3]. Detecting technological recombination, which can assist researchers in discovering potential technological innovation opportunities, exploring development trends of technologies, breaking through technical bottlenecks, and thus providing guidance for enterprise managers and technical policymakers [4].

* Corresponding Author

In recent years, researchers have attached great importance to recombination innovation, especially in the service and manufacturing industries [5]. For example, Corrocher et al. developed performance evaluation systems to measure the recombination innovation ability in service firms [6]. However, few researchers consider the recombination of technological domains [1], and the study on quantitatively measuring technological recombination opportunity is still limited [7].

In the field of scientific and technological innovation management, network analytics has been widely used to explore innovation activities, which can excavate the multi-dimensional relationship among knowledge elements [8]. The link prediction model is a network analysis-based method, which can determine the possibility of edges between unconnected nodes based on the topology information [9]. This method provides a novel perspective for exploring technological recombination opportunities, which could be applied to generate the probability of two technologies recombining in the future. However, recombinative innovation and technological advancement are continuous processes. With the emergence of new combinations among technologies, the technology network becomes highly dynamic [10]. Although many static link prediction methods have been developed, ignoring the time information and the dynamic evolution characteristics of the network over time, resulting in the inaccurate of the predicted network in dynamic network tasks [11]. Therefore, a future-oriented link prediction method based on dynamic network analysis should be presented to identify potential recombination opportunities.

Furthermore, semantic analysis has been introduced in technological innovation research to capture the semantic association between technologies [12]. The word embedding method is a text mining technology, which can effectively excavate underlying semantic and contextual relationships between keywords [13]. However, the static word embedding methods ignore the potential semantic changes of words in the dynamic context and the transformations of hidden semantic association patterns behind temporal keyword networks, which leads to the inaccurate description of the relationship between keywords, and reduces the accuracy of dynamic keyword network construction [14].

To address these concerns, we propose a novel framework for detecting technological recombination using semantic analysis and dynamic network analysis. The proposed method integrates a dynamic word embedding model, a dynamic network link prediction method, and machine learning technologies, which have the following three specific functions: 1) the dynamic network constructed by the temporal word embedding model to capture the changes of hidden semantic association patterns behind keyword networks over time and improve the accuracy of dynamic network construction; 2) the application of E-LSTM-D model comprehensively considering the network topological structure and time characteristics, and improving the accuracy of the predicted network; 3) the combination of SLM (smart local moving) community detection and PageRank algorithm to identify core keywords in each community, and then detect technological recombination candidates corresponding to the core keywords. We use a case study on artificial intelligence domain to demonstrate the reliability of our proposed method.

## 2 Method

The framework of detecting technological recombination is shown in Figure 1.

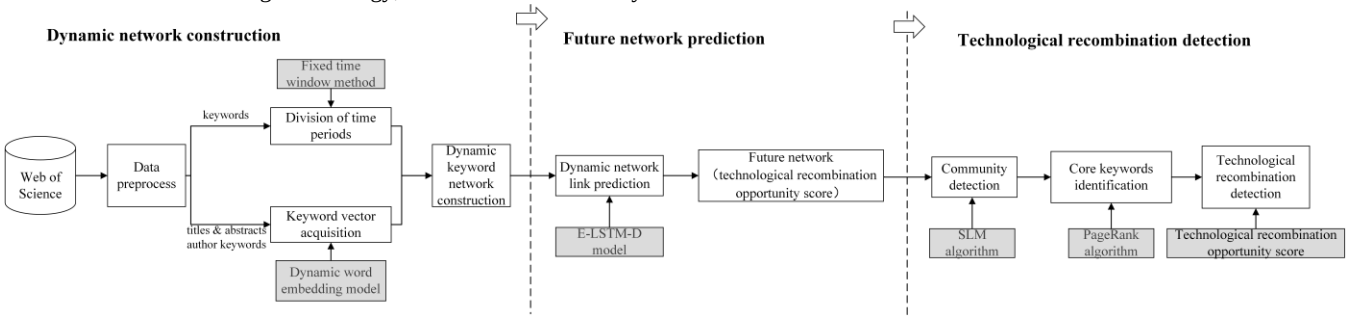


**Figure 1. Framework of detecting technological recombination**

### 2.1 Dynamic network construction

*2.1.1 Data collection and preprocessing*

The dataset gathered in this paper is acquired from the Web of Science (WoS) using a specific search strategy. The natural language processing function of VantagePoint (VP)[1] is used to extract terms from titles, abstracts, and author keywords, and the term clumping method is applied to process terms [15]. Following the theory of Arthur, we can understand the nature of technology and how it evolves [16]. This theory states that each technology is composed of some combination of components or principles and each component of the technology is a miniature technology. Therefore, the keywords (the research objects of our study) are the terms processed by the term clumping method.

*2.1.2 Word vector acquisition based on temporal word embedding model*

[1] https://www.thevantagepoint.com/.

The aim of this section is to generate the dynamic word vectors of keywords for each time slice by using temporal word embedding model. The temporal word embedding is a dynamic word embedding method [17], which has the following advantages: 1) solving the problem of polysemy in different contexts and improving the semantic accuracy of word representation; 2) capturing the transformation of hidden semantic association patterns in keyword networks over time and thus the accuracy of dynamic network construction can be improved; 3) processing a large number of corpus and words with a higher training speed and efficiency. The steps of generating word vectors are as follows.

(1) Division of time periods

The entire dataset, including text corpus and keywords, is divided based on the fixed time window method. It is a quasi-periodic time series division method that segments the text sequence into several slices with fixed length using the time window [18].

(2) Static word vector acquisition

We use Word2Vec model, which is trained through the entire dataset as a corpus, to generate static word vectors corresponding to all keywords and set them as the input of temporal word embedding model. Word2Vec is a static word embedding method that can capture the semantic information between keywords effectively [19]. In our study, skip-gram method is applied, which has been proven to have certain advantages in the research of the bibliometric domain [20]. The input is a text corpus including titles and abstracts in the whole dataset. Finally, the keywords are mapped into low dimensional and dense static word vectors that contain semantic information.

(3) Dynamic word vector acquisition

Then, the temporal word embedding model is applied to generate the dynamic embeddings with higher semantic accuracy on each time slice. In our study, the generation process of dynamic word vectors is as follows.

① PMI matrix generation

PMI (pointwise mutual information) matrix is generated on each time slice based on the text corpus. PMI is an information measurement index, which has been widely applied to measure the semantic similarity between words in the natural language processing tasks [21]. In our study, the PMI matrix is used to measure the correlation of any two keywords on each time slice within a time window L. The calculation method of $PMI_t(a,b)$ is shown in formula (1).

$$PMI_t(a,b) = log\left(\frac{C(a,b)\cdot|D_t|}{C(a)\cdot C(b)}\right) \quad (1)$$

Where $D_t$ represents the corpus on time slice t, $C(a,b)$ is the number of times that keyword a and b cooccur in corpus $D_t$ within a time window size L, $C(a)$ and $C(b)$ represent the number of occurrences of keyword a and b in $D_t$ in time slice t, respectively, and $|D_t|$ is the total number of keywords appear in $D_t$.

② PPMI matrix generation

Considering that the $PMI_t(a,b)$ will approach a large negative value when two keywords appear at the same time on the time slice with a low frequency in our corpus, which will lead to the unreasonable decomposition process of the matrix. Therefore, the PPMI (positive PMI) matrix is replaced with the PMI matrix to alleviate the data sparsity and make the model more stable [22]. The calculation method of $PPMI_t(a,b)$ is shown in formula (2).

$$PPMI_t(a,b) = max\{PMI_t(a,b), 0\} \quad (2)$$

Where $PMI_t(a,b)$ represents the correlation degree of keyword a and b in the PMI matrix on time slice t.

③ PPMI matrix factorization

The PPMI matrix is factorized by solving the optimization problem (3) to generate the keyword vectors corresponding to each time slice. The optimal solution of this problem is the dynamic word vector corresponding to the keyword of each time slice. The dynamic word vector set is shown in formula (4).

$$\min_{U_1,\cdots,U_T} \frac{1}{2}\sum_{t=1}^{T}||PPMI_t - U_1 U_T{}^T||_F^2 + \frac{\alpha}{2}\sum_{t=1}^{T}||U_t||_F^2 + \frac{\beta}{2}\sum_{t=2}^{T}||U_{t-1} - U_t||_F^2 \quad (3)$$

$$U = \{U_1, \cdots, U_T\} \quad (4)$$

Where $PPMI_t$ represents the PPMI matrix on time slice t, $U_t$ $(t = 1, \cdots, T)$ represents the word vector set on time slice t, T is the number of time slice, $||U_t||_F^2$ and $||U_{t-1} - U_t||_F^2$ represent the penalty term and smoothing normalization term, respectively, and $\alpha, \beta > 0$ represent the coefficients of penalty term and smoothing normalization term, respectively.

In this section, we finally generate the keyword vectors for each time slice.

*2.1.3 Dynamic network construction*

The purpose of this section is to construct a dynamic network based on the keyword vectors in each time slice. Firstly, the semantic similarity between keywords is measured using the cosine distance between keyword vectors. The calculation method of the semantic similarity $sim(a,b)$ between keyword a and b on time slice t is shown in formula (5).

$$sim(a,b) = \frac{U_t(a)^T U_t(b)}{||U_t(a)||_2 \cdot ||U_t(b)||_2} \quad (5)$$

Where $U_t(a)$ and $U_t(b)$ denote the dynamic word vectors of keyword a and b on time slice t respectively.

Then, the keyword network on each time slice is constructed based on the semantic similarity. In this paper, the keyword network on time slice t can be defined as $G_t = (V, E_t, W_t)$, where $V$, $E_t$, and $W_t$ denote the keywords, edges, and edge weights (semantic similarity) in $G_t$ respectively, $A_t$ is the adjacency matrix of $G_t$. Following the design of Zeng et al. [23], the similarity threshold filtering method is used to remove edges between the weak-related keyword pairs. When the edge weight is greater than the threshold, the edge between keywords is retained, otherwise, it is removed.

Finally, the keyword networks in all time slices constitute the dynamic network $G$, which can be represented as:

$$G = \{G_1, \cdots, G_t, \cdots, G_T\} \quad (6)$$

where $G_t$ denotes the keyword network on time slice t.

## 2.2 Future network prediction

After constructing the dynamic network, the dynamic network link prediction method is applied to predict the future network, and generate the technological recombination opportunity score, which represents the possibility of potential recombination between technologies. This part includes two sections: 1) Dynamic network link prediction model construction and 2) Future network prediction.

*2.2.1 Dynamic network link prediction model construction*

The Encoder-LSTM-Decoder (E-LSTM-D) model, which is a dynamic network link prediction method based on deep learning, is applied to predict the future network. The E-LSTM-D model showed its advantages in the dynamic network analysis [24]: 1) it can be suitable for different scale networks since the encoder-decoder architecture can deal with high-dimensional, nonlinear, and data-sparse networks effectively; 2) the stacked LSTM structure can capture richer time information and better learn network topology characteristics and dynamic evolution patterns. The steps of constructing the dynamic network link prediction are as follows.

(1) Network data modeling

The dynamic network G is modeled as a series of graphic sequences with fixed length and time interval. In each graphic sequence, the future network is predicted based on the historical networks. Next, the training set and test set are proportionally divided, where the training set is used to train and generate the link prediction model, and the test set is used to evaluate the performance of the model based on indicators [10]. To evaluate the performance of our trained link prediction model, we choose three indicators: AUC [25], Precision [26], and Error rate [24]. The Error rate is a good supplement to AUC and can comprehensively measure the performance of the dynamic network link prediction method.

(2) Dynamic network link prediction model training

Firstly, the encoder-decoder architecture and the stacked LSTM structure are built. Then, forward propagation is used to obtain the loss, followed by back propagation to update all parameters, including the weight parameters and deviation parameters of the encoder layer, decoder layer, and LSTM structure. When the loss function is minimized, the trained link prediction model is generated.

In this section, the final model is generated by testing the parameters constantly, including the number of encoder layers, decoder layers, LSTM modules, and their neural units. Then, the trained dynamic network link prediction model is used for predicting the future network.

*2.2.2 Future network prediction*

The future network $G_{T+1}$ is predicted based on the latest networks $\{G_{T-N+1}, G_{T-N+2}, \cdots, G_T\}$, which provides the basis for the technological recombination opportunity analysis. Since the latest network can better reflect the current development tendency of technologies, the well-trained model is applied in $\{G_{T-N+1}, G_{T-N+2}, \cdots, G_T\}$ to calculate the possibility of the connection between keywords in the future network. We define the possibility of keyword pairs to be connected as the technological recombination opportunity score, which denotes the possibility of potential recombination between technologies.

Then, the technological recombination opportunity score is transformed into the edge weight in the future network. Following the study of Wang et al. [27], the weight threshold is set, and the technological recombination opportunity score greater than this threshold would be converted into the edge weight. In our study, the future network can be defined as $G_{T+1} = (V, E_{T+1}, W_{T+1})$, where $V$, $E_{T+1}$, and $W_{T+1}$ denote the keywords, edges, and edge weights (technological recombination opportunity score) in $G_{T+1}$ respectively, $A_{T+1}$ is the adjacency matrix of $G_{T+1}$.

In this section, we generate the future network $G_{T+1}$ containing the technological recombination opportunity score (edge weight $W_{T+1}$).

## 2.3 Technological recombination detection

This part aims to analyze the potential technological recombination opportunities based on the technological recombination opportunity scores between technologies in the future network.

*2.3.1 Community detection based on SLM*

After constructing the future network, SLM (smart local moving) community detection is introduced to cluster keywords into communities. The identified community contains multiple keywords that are closely related to each other. Keywords within the same community are more likely to produce recombinative innovation than those within different communities. Therefore, potential technology recombination opportunities can be better discovered in the community.

SLM is a modularity-based community detection algorithm, which can achieve high-quality division results in large-scale networks [28]. Modularity was proposed by Newman and Girvan in 2004, which is an index to measure the quality of the detected community [29]. The greater the modularity, the better the performance of community division. In our study, the modularity can be calculated as:

$$M = \frac{1}{2L}\sum_{i,j}[W_{ij} - \frac{k_i k_j}{2L}]\delta(c_i, c_j) \quad (7)$$

where L is the total number of edges in the future network $G_{T+1}$, $k_i$ and $k_j$ denote the sum of the edge weights of keyword i and j, respectively, $W_{ij}$ is the edge weight between keyword i and j. $c_i$ and $c_j$ represent the community number which the keyword i and j belong to, respectively, if $c_i = c_j$, then $\delta(c_i, c_j) = 1$, otherwise, $\delta(c_i, c_j) = 0$.

In our study, the i-th community generated from community detection is defined as $C_i = (V_i, E_i, W_i)$, where $V_i$, $E_i$, and $W_i$ denote the keywords, edges, and edge weights (technological recombination opportunity score) in $C_i$ respectively, $A_i$ is the adjacency matrix of $C_i$. Finally, the community set $C$ is generated in the future network using SLM algorithm, which is represented as:

$$C = \{C_1, \cdots, C_i, \cdots, C_s\} \quad (8)$$

where $C_i$ denotes the i-th community in the future network, and s is the number of communities that has been detected.

*2.3.2 Core keywords identification based on PageRank algorithm*

In this section PageRank algorithm is applied to measure the importance score of keywords and thus identify the core keywords in the communities. This algorithm fully considers multiple factors including the local topological structure of the target node and the importance of the nodes connected with it [30], which has been widely introduced to identify core nodes in various complex networks [31]. Therefore, this method is introduced to rank the importance of keywords in each community. The importance score $PR(i)$ is calculated as:

$$PR(i) = d \times \sum_{j=1}^{n} \frac{PR(T_j)}{C(T_j)} + (1 - d) \quad (9)$$

where d is the damping factor ($0 \leq d \leq 1$), generally 0.85, $T_j$ denotes the keyword linked to the keyword $i$, $C(T_j)$ is the number of keywords linked with $T_j$, and $n$ is the number of keywords linked with keyword $i$.

Finally, we sort all keywords in the community based on the importance score, and select the top-K keywords as the core keywords in each community.

*2.3.3 Potential technological recombination detection*

After identifying core keywords in communities, technological recombination candidates corresponding to core keywords are detected based on the technological recombination opportunity score, which reveals the recombinative innovation among technologies. This paper only considers whether the technologies' new combination can produce recombinative innovation, provide researchers with solutions for difficult problems in a certain field and explore development trends of technologies in the future.

Firstly, the predicted future network $G_{T+1}$ is compared with the current network $G_T$ to select new edges in each community and acquire the edge weights (technological recombination opportunity score) corresponding to the new edges. Then, according to the edge weights of the new edges, the top-3 technologies (keywords) are identified as the technological recombination candidate corresponding to the target keyword, revealing the potential recombinative innovation corresponding to the core keywords.

# 3 Case study

As an emerging field of multidisciplinary research and innovation, artificial intelligence (AI) has a broad prospect of continuous development, which offers potential opportunities for detecting technological recombination within this field. Given that AI is at the nascent stage of development and immature, we chose AI as the field to explore more cross integration directions, and verify the effectiveness of our framework.

## 3.1 Dynamic network construction

Following the study by Liu et al. [32], we acquired 240561 papers between 2014 to 2020 from the Web of Science (WoS). The search strategy used in this paper is shown in "Appendix A". Then, VantagePoint (VP) and the term clumping method were used to process keywords and text corpus. Finally, a total of 11773 keywords remained.

The experimental environment is Windows 10 operating system, the method is implemented using Python 3.7, mainly using dependency module genism, tensorflow, keras, pandas, and network.

Then, a fixed time window method was introduced to divide the time periods, in which the time window was set to 1 year. The whole dataset was divided into seven time slices, of which the time slice of 2020 was used for validation analysis.

Following the design in Section 2.1.2, the Word2Vec model was applied to generate static keyword vectors as the initial value of dynamic embedding vectors, in which the entire data set was used as a corpus. We set the word vector dimension to 100 and the window size to 5. Next, the temporary word embedding model was applied to generate the word vectors corresponding to the keywords on each time slice and set the parameter α=10 and β=50.

Next, the semantic similarity between keywords was measured according to formula (5), and then similarity matrixes were generated. The partial similarity matrix of 2019 is shown in Table 1.

**Table 1. Partial similarity matrix of 2019**

| | support vector machine | convolutional neural network | random forest | decision tree learning | particle swarm optimization |
|---|---|---|---|---|---|
| support vector machine | 1 | 0.7488 | 0.9385 | 0.9025 | 0.8743 |
| convolutional neural network | 0.7488 | 1 | 0.7445 | 0.7953 | 0.7195 |
| random forest | 0.9385 | 0.7445 | 1 | 0.8715 | 0.7978 |
| decision tree learning | 0.9025 | 0.7953 | 0.8715 | 1 | 0.8843 |
| particle swarm optimization | 0.8743 | 0.7195 | 0.7978 | 0.8843 | 1 |

Finally, similarity matrixes were transformed into keyword networks, and the keyword networks on six time slices constitute the dynamic network.

## 3.2 Future network prediction

After constructing the dynamic network, a future network was predicted subsequently. Firstly, we set 3 years as a length and 1 year as a sliding window to generate the graphic sequences. For each graphic sequence, the third network is predicted based on the first two networks. Four graphic sequences were acquired from the dynamic network in this paper. Table 2 shows the details of data division. In our study, the first three graphic sequences were set as the training set, and the last graphic sequence was the test set.

**Table 2. Dataset division**

| graphic sequence | year |
|---|---|
| 1 | 2014-2016 |
| 2 | 2015-2017 |
| 3 | 2016-2018 |
| 4 | 2017-2019 |

Then, the well-trained dynamic network link prediction model was generated through a tuning process of parameters. The parameter configuration of our trained model is given in Table 3.

**Table 3. Parameter configuration of our trained model**

| Parameters | Values |
|---|---|
| No. units in encoder | 1024\|512 |
| No. units in stacked LSTM | 384\|384 |
| No. units in decoder | 512\|11773 |
| Learning rate | 0.001 |
| Weight decay factor | 5e-4 |

After generating the model, the test set was inputted into the model to calculate the values of evaluation indicators (AUC, Precision, and Error rate). The AUC value is 0.902, the Precision value is 0.896, and the Error rate value is 0.960 in the trained model, which indicates that our trained model has a good performance.

Following Section 2.2.2, we applied the trained model to the 2018 and 2019 networks and acquired the possibility of the connection between keywords in the predicted network, that is, the technological recombination opportunity score. With the weight threshold set as 0.9, we finally generated a future network with 2657691 edges and a network density of 0.038.

## 3.3 Technological recombination detection

The next stage was to detect the potential technological recombination. Following Section 2.3.1, the SLM community detection algorithm was applied to divide the predicted network and 11773 keywords were clustered into 42 communities. We numbered each community according to the total number of keywords in each community. That is the number of keywords in communities 1 to 42 decrease sequentially.

Next, the PageRank algorithm was used to calculate the important scores of keywords and thus identify the core keywords in 42 communities, and we summarized the main research topics in the community (community description) according to the core keywords. Table 4 lists the details of top-5 core keywords contained in the community 1-5. The hot topics in the artificial intelligence domain can be explored according to the core keywords.

**Table 4. Top-5 core keywords within the community 1-5**

| No | Top-5 core keywords | The description of the community | No. keywords in community |
|---|---|---|---|
| 1 | reinforcement learning process, response functions, image coding, graph topology, posterior probability distribution | Reinforcement learning | 459 |
| 2 | semantic mapping, facial landmark detection, image inpainting, manifold alignment, hypergraph learning | semantic comprehension | 449 |
| 3 | heuristic search, hyperspectral band selection, functional data analysis, statistical hypothesis testing, stochastic learning | stochastic search | 440 |
| 4 | sequential forward selection, grid search algorithm, hybrid learning algorithm, backward elimination, | search algorithm | 414 |

| | | | |
|---|---|---|---|
| | bootstrap method | | |
| 5 | medical domain, clinical domain, social media, biological research, emotion analysis | medical domain | 399 |

According to the results of the PageRank algorithm, the hot topics in artificial intelligence domain mainly include reinforcement learning, semantic comprehension, stochastic search, feature representation, multi-objective analysis, visual detection, and intelligent algorithm. The application fields mainly comprise medical science, ecosystem, social discovery, behavioral gene, and mechanical system. Meanwhile, the researchers in this field attach great importance to the evaluation and verification of methods.

Following the design in Section 2.3.3, the top-3 technologies (keywords) were selected as the technological recombination candidates corresponding to the core keywords within the community, based on the edge weight (technological recombination opportunity score) of the new edge. The core keywords (top-5) in community 1 and the technological reorganization candidates are given in Table 5.

**Table 5. Potential technological recombination**

| No | Core keywords | Important score | Technological recombination candidates | Technological recombination opportunity score |
|---|---|---|---|---|
| 1 | Reinforcement learning process | 0.00677 | spatial properties | 0.9990 |
| | | | system configuration | 0.9980 |
| | | | target location | 0.9975 |
| 2 | response functions | 0.00619 | video cameras | 0.9999 |
| | | | DFT calculations | 0.9956 |
| | | | reinforcement learning process | 0.9941 |
| 3 | image coding | 0.00504 | Markov processes | 0.9999 |
| | | | probabilistic learning | 0.9998 |
| | | | confidence value | 0.9914 |
| 4 | graph topology | 0.00499 | state estimates | 0.9997 |
| | | | Markov processes | 0.9973 |
| | | | reinforcement learning process | 0.9961 |
| 5 | posterior probability distribution | 0.00483 | noise distribution | 0.9930 |
| | | | response functions | 0.9921 |
| | | | point estimates | 0.9919 |

Several observations can be acquired based on the above results. There are correlations between keywords in the technological reorganization candidates corresponding to each core keyword, and they all have great correlations with the target keywords, which demonstrates that the results of technological recombination identified are reliable.

Two main patterns of potential technological recombination are identified in this paper: 1) a new combination of two previous technologies; 2) expansion of technological application scenarios.

We further discuss the essence meaning between core keywords and their technological reorganization candidates. For example, the technological recombination candidates corresponding to the core keyword "reinforcement learning process" are "spatial properties", "system configuration", and "target location". In recent years, reinforcement learning is regarded as an intelligent search algorithm in the navigation domain, considering more space-time characteristics, and proposing the solution of target location, system configuration

optimization, and location search. Therefore, the recombination of core keywords with its technological recombination candidate can generate innovation and inspire new ideas.

## 3.4 Validation

In this part, we conducted the quantitative and qualitative methods to verify the reliability of our proposed method and technological recombination detection results.

*3.4.1 Verification of the trained model and prediction results*

The performance of the link prediction model trained in this paper was validated based on AUC, Precision, and Error rate by comparing with six traditional link prediction methods. Baseline methods include Adamic-Adar-based method [33], Preferential attachment-based method [34], Jaccard-based method [35], Spectral Clustering-based method [36], Node2Vec-based method [37], and Variational Graph Auto-Encoders (VGAE) [38]. The comparison results are given in Table 6.

**Table 6. The comparison of prediction performance**

| **Methods** | | **AUC** | **Precision** | **Error rate** |
|---|---|---|---|---|
| Dynamic network link prediction method (our method) | E-LSTM-D | 0.902 | 0.896 | 0.960 |
| Traditional link prediction method | AA | 0.613 | 0.712 | 2.961 |
| | PA | 0.701 | 0.760 | 2.944 |
| | Jaccard | 0.507 | 0.600 | 2.999 |
| | SC | 0.556 | 0.642 | 3.002 |
| | Node2Vec | 0.686 | 0.797 | 2.993 |
| | VGAE | 0.685 | 0.747 | 2.888 |

It can be seen that our method outperforms baseline methods in three evaluation indicators. Concretely, compared with the six traditional link prediction methods, the AUC value of our method increases by 28.9%, 20.1%, 39.5%, 34.6%, 21.6%, and 21.7%, respectively, the Precision value increases by 18.4%, 13.6%, 29.6%, 25.4%, 9.9%, and 14.9%, respectively, and the Error rate value decreases by 2.001, 1.984, 2.039, 2.042, 2.033, and 1.928, respectively. These results demonstrate the dynamic network link prediction model used in this paper has achieved good performance on our dataset.

Furthermore, the accuracy of the prediction results has been verified via the comparison between the future network and the 2020 real keyword network. Specifically, the network of 2020 was constructed, including 11773 nodes and 3396060 edges, with a network density of 0.049. In the predicted future network, there are 1637788 links with a connection possibility greater than 0.95, of which 1416686 edges appear in the 2020 real network, so the calculated Precision value is 0.865. The results indicate that our trained dynamic network link prediction model can provide reliable results.

*3.4.2 Verification of detected technological recombination*

In this section, the qualitative method was applied to verify the reliability of the technological recombination detection results by searching relevant articles, patents, and other literature. Note that the published time of literature should be limited to 2020 and beyond. Table 7 shows the detailed empirical evidence of partial potential technological recombination.

**Table 7. Relevant documentary proof of partial potential technological recombination**

| **No** | **Core keywords** | **Potential technological recombination candidates** | **Relevant documentary proof** |
|---|---|---|---|
| 1 | reinforcement learning process | spatial properties | Wang et al. recombined reinforcement learning and spatial properties, which solved the problem of sensor layout optimization in 2020 [39]. |

| | | | |
|---|---|---|---|
| | | system configuration | Wee and Nayak recombined reinforcement learning and system configuration, which solved the problem of data replication system configuration optimization in the IT environment in 2020 [40]. |
| | | target location | Song et al. recombined reinforcement learning and target location, which improved the accuracy of target positioning in 2020 [41]. |
| 2 | semantic mapping | transfer learning algorithm | Hou et al. recombined semantic mapping and transfer learning algorithm, which improved the accuracy of semantic mapping in 2020 [42]. |
| | | person re-identification | Zhao and Xu recombined semantic mapping and person re-identification, which improved the accuracy of the person re-identification task in 2020 [43]. |
| | | medical images | Li et al. proposed a novel method to recombine semantic mapping and medical images, which solved the problem of semantic segmentation of medical images in 2021. [44]. |
| 3 | heuristic search | interval analysis | Purini et al. recombined heuristic search and interval analysis to overcome the shortcomings of traditional distance analysis technology in 2020 [45]. |
| | | grey system theory | Yao et al. recombined heuristic search and grey system theory, which solved the problem of test selection in 2020 [46]. |
| | | stochastic learning | NESI et al. recombined heuristic search algorithm and stochastic learning, which proposed a novel super heuristic algorithm for acquiring, storing, and retrieving heuristic knowledge in 2020 [47]. |

Table 7 demonstrates the alignment between our technological recombination results detected and the literature. Therefore, the potential technological recombination identified in this paper is reliable, and the effectiveness of the proposed method has been further verified.

## 4 Conclusion

In this paper, we proposed a novel methodology to detect technological recombination using semantic analysis and dynamic network analysis. Temporal word embedding model was applied to construct the dynamic keyword network, capturing the changes of hidden semantic association modes in different keyword networks effectively, and improving the accuracy of the dynamic network construction. Further, the E-LSTM-D method based on the dynamic network analysis was combined with temporal word embedding model to predict the future network, exploring the dynamic evolution characteristics of the keyword network over time and improving the performance of potential technological recombination detection.

Semantic analysis and dynamic network analysis were combined to identify potential technological reorganization, which provides technical intelligence on recombinative innovation. In addition, this method can not only identify the recombination innovation between technologies, but also explore the potential development trend of technologies and predict the research hot topics in the field of artificial intelligence in the future.

Several limitations of our proposed method require further improvement: 1) The data is only limited to the period from 2014 to 2020, reducing the accuracy of network prediction. More abundant dataset should be further used for dynamic network link prediction research; 2) This paper does not provide a more detailed classification of technologies. In the future, the conceptual model should be considered to categorize the technologies in detail, making the research of technological recombination play a greater value; 3) More latest algorithms with good performance would be tried in our study, such as TextRank.

## ACKNOWLEDGMENTS

This work was supported by the National Science Foundation of China Funds [Grant No. 71774013].

## REFERENCES

[1] Guan, J. C., & Yan, Y. Technological proximity and recombinative innovation in the alternative energy field. Research Policy, 45(7): 1460-1473, 2016.

[2] Gruber, M., Harhoff, D., & Hoisl, K. Knowledge recombination across technological boundaries: Scientists vs. engineers. Management Science, 59(4): 837-851, 2013.

[3] Ramesh, A., Pavlov, M., Goh, G., & Gray, S., et al. Zero-shot text-to-image generation. In International Conference on Machine Learning (PMLR), pages 8821-8831, 2021.

[4] Corredoira, R. A., & Banerjee, P. M. Measuring patent's influence on technological evolution: A study of knowledge spanning and subsequent inventive activity. Research Policy, 44(2), 508-521, 2015.

[5] Bessant, J., & Trifilova, A. Developing absorptive capacity for recombinant innovation. Business Process Management, 2017.

[6] Corrocher, N., & Zirulia, L. Demand and innovation in services: The case of mobile communications. Research Policy, 39(7): 945-955, 2010.

[7] Zhou, X., Huang, L., Zhang, Y., & Yu, M. A hybrid approach to detecting technological recombination based on text mining and patent network analysis. Scientometrics, 121(2): 699-737, 2019.

[8] Liu, Z., Yin, Y., Liu, W., & Dunford, M. Visualizing the intellectual structure and evolution of innovation systems research: a bibliometric analysis. Scientometrics, 103(1), 135-158, 2015.

[9] Kumar, A., Singh, S. S., Singh, K., & Biswas, B. Link prediction techniques, applications, and performance: A survey. Physica A: Statistical Mechanics and its Applications, 553, 124289, 2020.
[10] Huang, L., Chen, X., Ni, X., Liu, J., Cao, X., & Wang, C. Tracking the dynamics of co-word networks for emerging topic identification. Technological Forecasting and Social Change, 170: 120944, 2021.
[11] Huang, L., Chen, X., Zhang, Y., Zhu, Y., Li, S., & Ni, X. Dynamic network analytics for recommending scientific collaborators. Scientometrics, 126(11): 8789-8814, 2021.
[12] Wang, Z., Ma, L., & Zhang, Y. A hybrid document feature extraction method using latent Dirichlet allocation and word2vec. In 2016 IEEE first international conference on data science in cyberspace (DSC) (pp. 98-103). 2016.
[13] Pennington, J., Socher, R., & Manning, C. D. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pages 1532-1543, 2014.
[14] Wang, Y., Hou, Y., Che, W., & Liu, T. From static to dynamic word representations: a survey. International Journal of Machine Learning and Cybernetics, 11(7): 1611-1630, 2020.
[15] Zhang, Y., Porter, A. L., Hu, Z., Guo, Y., & Newman, N. C. "Term clumping" for technical intelligence: A case study on dye-sensitized solar cells. Technological Forecasting and Social Change, 85: 26-39, 2014.
[16] Arthur, W. B. The nature of technology: What it is and how it evolves. Simon and Schuster, 2009.
[17] Yao, Z., Sun, Y., Ding, W., Rao, N., & Xiong, H. Dynamic word embeddings for evolving semantic discovery. In Proceedings of the eleventh acm international conference on web search and data mining, pages 673-681, 2018.
[18] Sheng, Z., Hailong, C., Chuan, J., & Shaojun, Z. An adaptive time window method for human activity recognition. In 2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE), pages 1188-1192, 2015.
[19] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, 26, 2013.
[20] Zhang, C., Huang, C., Yu, L., Zhang, X., & Chawla, N. V. Camel: Content-aware and meta-path augmented metric learning for author identification. In Proceedings of the 2018 World Wide Web Conference, pages 709-718, 2018.
[21] Su, Q., Xiang, K., Wang, H., Sun, B., & Yu, S. Using pointwise mutual information to identify implicit features in customer reviews. In International Conference on Computer Processing of Oriental Languages, pages 22-30, 2006.
[22] Levy, O., & Goldberg, Y. Neural word embedding as implicit matrix factorization. Advances in neural information processing systems, 27, 2014.
[23] Zeng, Q., Hu, X., & Li, C. Extracting keywords with topic embedding and network structure analysis. Data Analysis and Knowledge Discovery, 3(7): 52-60, 2019.
[24] Chen, J., Zhang, J., Xu, X., Fu, C., Zhang, D., Zhang, Q., & Xuan, Q. E-lstm-d: A deep learning framework for dynamic network link prediction. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 51(6): 3699-3712, 2019.
[25] Hanley, J. A., & McNeil, B. J. The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology, 143(1): 29-36, 1982.
[26] Lü, L., & Zhou, T. Link prediction in complex networks: A survey. Physica A: statistical mechanics and its applications, 390(6): 1150-1170, 2011.
[27] Wang, W., Li, X., & Yu, S. Chinese Text Keyword Extraction Based on Doc2vec And TextRank. In 2020 Chinese Control And Decision Conference (CCDC), pages 369-373, 2020.
[28] Waltman, L., & Van Eck, N. J. A smart local moving algorithm for large-scale modularity-based community detection. The European physical journal B, 86(11): 1-14, 2013.
[29] Newman, M. E., & Girvan, M. Finding and evaluating community structure in networks. Physical review E, 69(2): 026113, 2004.
[30] Marjai, P., & Kiss, A. Influential Performance of Nodes Identified by Relative Entropy in Dynamic Networks. Vietnam Journal of Computer Science, 8(1): 93-112, 2021.
[31] LIANG, T., LI, C., & LI, H. Top-k Learning Resource Matching Recommendation Based on Content Filtering PageRank. Computer Engineering, 43(2): 220-226, 2017.
[32] Liu, N., Shapira, P., & Yue, X. Tracking developments in artificial intelligence research: constructing and applying a new search strategy. Scientometrics, 126(4): 3153-3192, 2021.
[33] Adamic, L. A., & Adar, E. Friends and neighbors on the web. Social networks, 25(3): 211-230, 2003.
[34] Zhou, T., Lü, L., & Zhang, Y. C. Predicting missing links via local information. The European Physical Journal B, 71(4): 623-630, 2009.
[35] Liben‐Nowell, D., & Kleinberg, J. The link‐prediction problem for social networks. Journal of the American society for information Science and technology, 58(7): 1019-1031, 2007.
[36] Tang, L., & Liu, H. Leveraging social media networks for classification. Data Mining and Knowledge Discovery, 23(3), 447-478 2011.
[37] Grover, A., & Leskovec, J. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, pages 855-864, 2016.
[38] Kipf, T. N., & Welling, M. Variational graph auto-encoders. Ar Xiv preprintar Xiv:Conference Name:ACM Woodstock, 1611: 07308, 2016.
[39] Wang, Z., Li, H. X., & Chen, C. Reinforcement learning-based optimal sensor placement for spatiotemporal modeling. IEEE transactions on cybernetics, 50(6): 2861-2871, 2019.
[40] Wee, C. K., & Nayak, R. Adaptive Data Replication Optimization Based on Reinforcement Learning. In 2020 IEEE Symposium Series on Computational Intelligence EEE, pages 1210-1217, 2020.
[41] Song, K., Zhang, W., Lu, W., Zha, Z. J., Ji, X., & Li, Y. Visual object tracking via guessing and matching. IEEE Transactions on Circuits and Systems for Video Technology, 30(11): 4182-4191, 2019.
[42] Hou, C., Zhao, X., & Lin, Y. Depth Estimation and Object Detection for Monocular Semantic SLAM Using Deep Convolutional Network. In 2020 IEEE 20th International Conference on Software Quality, Reliability and Security Companion (QRS-C), pages 256-263, 2020.
[43] Zhao, X., & Xu, X. Multi-granularity and Multi-semantic Model for Person Re-identification in Variable Illumination. In 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pages 3154-3161, 2020.
[44] Li, S., Gao, Z., & He, X. Superpixel-Guided Iterative Learning from Noisy Labels for Medical Image Segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 525-535, 2021.
[45] Purini, S., Benara, V., Choudhury, Z., & Bondhugula, U. Bitwidth customization in image processing pipelines using interval analysis and smt solvers. In Proceedings of the 29th International Conference on Compiler Construction, pages 167-178, 2020.
[46] Yao, Z., Zhu, L., Zhang, T., & Wang, J. Optimal Selection of Tests for Fault Diagnosis in Multi-Path System with Time-delay. Journal of Electronic Testing, 36(1): 75-86, 2020.
[47] Nesi, L. C., & da Rosa Righi, R. H2-SLAN: A hyper-heuristic based on stochastic learning automata network for obtaining, storing, and retrieving heuristic knowledge. Expert Systems with Applications, 153: 113426, 2020.

## Appendix A. Search strategy for artificial intelligence

This appendix shows the search strategy of papers for artificial intelligence used in this paper: TS=("Artifcial Intelligen*" or "Neural Net*" or "Machine* Learning" or "Expert System$" or "Natural Language Processing" or "Deep Learning" or "Reinforcement Learning" or "Learning Algorithm$" or "*Supervised Learning" or "Intelligent Agent*"or ("Backpropagation Learning" or "Back-propagation Learning" or "Bp Learning") or ("Backpropagation Algorithm*" or "Back-propagation Algorithm*") or "Long Short-term Memory" or ((Pcnn$ not Pcnnt) or "Pulse Coupled Neural Net*") or "Perceptron$" or ("Neuro-evolution" or Neuroevolution) or "Liquid State Machine*" or "Deep Belief Net*" or ("Radial Basis Function Net*" or Rbfnn* or"Rbf Net*") or "Deep Net*" or Autoencoder* or "Committee Machine*" or "Training Algorithm$" or ("Backpropagation Net*" or "Back-propagation Net*" or "Bp Network*") or "Q learning" or "Convolution* Net*" or "Actor-critic Algorithm$" or ("Feedforward Net*" or "Feed-

Forward Net*”) or“Hopfeld Net*” or Neocognitron* or Xgboost* or “Boltzmann Machine*” or “Activation Function$” or (“Neurodynamic Programming” or “Neuro dynamic Programming”) or “Learning Model*” or (Neurocomputing or “Neuro-Computing”) or “Temporal Diference Learning” or “Echo State* Net*”or “Transfer Learning” or “Gradient Boosting” or “Adversarial Learning” or “Feature Learning” or“Generative Adversarial Net*” or “Representation Learning” or (“Multiagent Learning” or “Multi-agent Learning”) or “Reservoir Computing” or “Co-training” or (“Pac Learning” or “Probabl* Approximate* Correct Learning”) or “Extreme Learning Machine*” or “Ensemble Learning” or “Machine* Intelligen*” or (“Neuro fuzzy” or Neurofuzzy) or “Lazy Learning” or (“Multi* instance Learning” or “Multiinstance Learning”) or (“Multi* task Learning” or “Multitask Learning”) or “Computation* Intelligen*” or “Neural Model*” or (“Multi* label Learning” or “Multilabel Learning”) or “Similarity Learning” or “Statistical Relation* Learning” or “Support* Vector* Regression” or “Manifold Regulari?ation” or “Decision Forest*” or “Generali?ation Error*” or “Transductive Learning” or (Neurorobotic* or“Neuro-robotic*”) or “Inductive Logic Programming” or “Natural Language Understanding” or (Adaboost* or “Adaptive Boosting”) or “Incremental Learning” or “Random Forest*” or “Metric Learning” or “Neural Gas” or “Grammatical Inference” or “Support* Vector* Machine*” or (“Multi* label Classifcation” or “Multilabel Classifcation”) or “Conditional Random Field*” or (“Multi* class Classifcation” or “Multiclass Classifcation”) or “Mixture Of Expert*” or “Concept* Drift” or “Genetic Programming” or “String Kernel*” or (“Learning To Rank*” or “Machine-learned Ranking”) or “Boosting Algorithm$” or “Robot* Learning” or “Relevance Vector* Machine*” or Connectionis* or (“Multi* Kernel$ Learning” or “Multikernel$ Learning”) or “Graph Learning” or “Naive bayes* Classif*” or“Rule-based System$” or “Classifcation Algorithm*” or “Graph* Kernel*” or “Rule* induction” or“Manifold Learning” or “Label Propagation” or “Hypergraph* Learning” or “One class Classif*” or “Intelligent Algorithm*”) OR WC=(“Artifcial Intelligence”).