2022 4th International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI) | 979-8-3503-3394-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/MLBDBI58171.2022.00060

Prediction of New Technical Terms Based on Machine Learning Algorithms

Haiying Ren*[a], Luyao Zhang[b]
[a,b]School of Economics and Management, Beijing University of Technology, Beijing, China
*renhaiying@bjut.edu.cn

ABSTRACT **New technical terms (NTTs), the precursors of technical terminology, are vehicles for new ideas contained in innovative technologies. Term identification research tends to mature, but little research has been done on the prediction of NTTs, which mines promising technical terms from voluminous technological data to help R&D teams rapidly form new ideas. In this study, we design predictor variables by constructing a prior co-word network and construct multiple machine learning models that predict the formation of NTTs. To validate the above approach, this paper focuses on the formation of some NTTs in the neural network domain. The prediction model constructed by the stacking model produced the best results with a prediction accuracy of 78.6%.**

Keywords: Patent analysis; Ideation process; New technical terms; Machine learning

I. INTRODUCTION

Advanced technology and product innovation are booming at a high speed, forming a torrent of the times. To stand firm in the torrent, we need to propose innovative ideas as early as possible to seize the innovation opportunities. However, in the research & development (R&D) process, the "fuzzy front end" (which describes the fuzziness in ideation) has always been a problem that R&D personnel struggles. In the existing research on new product development and innovation theory, the dominant view is that new ideas are novel combinations of existing ideas [1, 2], coming from multiple internal or external sources. However, in the era of big data, uncovering valuable information from massive amounts of technical data is becoming more difficult because of numerous such idea combinations.

Since new ideas are expressed through specific combinations of concepts and special terminology, the generation of new ideas is closely linked to the formation of new technical terms (NTTs). If we can discover and select the most potential NTTs from large volume of technical data, we would obtain a "vocabulary of the future" and generate new ideas with efficiency.

In the present work we are studying the above "NTT prediction problem". Predicting the NTTs will not only assist R&D teams in generating novel ideas for new product development and improving existing technologies, but also would answer, in part, the question of "where do the new ideas come from", therefore providing a new perspective on the "fuzzy front end" problem. In summary, the prediction of NTTs has significant theoretical and practical implications.

There are many studies on term identification, and some scholars use statistical methods to measure the unithood [3, 4] and termhood [5, 6] of phrases to identify *existing* terms. With the development of machine learning, studies on automatic identification of terms using machine learning have started to appear (e.g., [7]). However, the research on *new* technical terms is rarely mentioned by scholars.

Inspired by the fast development of big data analytics, we believe that the NTTs are not random word combinations. Instead, they have "valuable" word patterns and can be mined from historical technical data. Machine learning may be used to uncover the patterns of NTTs and predict them. In this vein, we construct prior co-word networks for NTTs, collect variables related to the formation of NTTs, and generate predictive models for NTTs using various machine learning methods. Since neural networks are widely used in various fields of intelligence technology with remarkable results [8], the proposed method is applied and validated in a the domain "computer systems using neural network models."

II. METHODOLOGY AND PROCEDURE

A. Obtain sample patents

Patent data were obtained from the Derwent patent database, and the search field patents corresponding to IPC="G06N-003/02" ("computer systems using neural network models"). The total of 230 patent records were randomly sampled from patents in this field between 2001 and 2015. Since the abstract is the part containing the key information of a patent, we selected the abstracts of these patents as the object of study. Afterwards, these abstracts were cleaned and pre-processed to exclude irrelevant information.

B. Extract NTTs from sample patents and generate non-technical terms

First, we used the SAO technique [9] to extract SO structures (SO is essentially a noun phrase in subject and object based on syntactic dependencies, or an NP in subject and object extracted from syntactic dependencies) from the sample abstracts. The extracted SO structures still contained some non-essential data, known as stop words. The data had been removed. We excluded excluding the two-word phrases in this study, for reasons to be given in the following Subsection D. Since NTTs are noun phrases that appear for the first time in a patent (such patents are called "focal patents" thereafter). Therefore, we searched the DII database to see if the extracted phrase appeared in "prior patents" (all patents filed before the

979-8-3503-3394-7/22/$31.00 ©2022 IEEE
DOI 10.1109/MLBDBI58171.2022.00060

Authorized licensed use limited to: VIT University. Downloaded on June 14,2024 at 06:05:56 UTC from IEEE Xplore. Restrictions apply.

priority year of the focal patent), identified phrases that did not appear in "prior patents" as NTTs. For example, “color value table” first appeared in US2006120596-A1 in DII, and was considered an NTT.

To better analyze the formation of NTTs and make predictions, we used NTTs as a positive sample and constructed the same number of non-technical terms (notTTs) as a negative sample for comparison with NTTs. Like NTTs, notTTs were also considered to be a combination of random words before they appeared, so we constructed the notTT sample with a batch of random word combinations from the focal patents that have so far not appeared in any among the patents. They may be a random combination of nouns that did not conform to grammatical rules, or they may be a technical phrase whose meaning had not yet been discovered. For example, the random word combination “hidden probability back” had not appeared in any patent in DII so far, and was considered a notTT. To ensure the model was as fair as possible, we constructed the same proportion of 3-word, 4-word, and 5-word notTTs as NTTs (the number of NTTs with more than 5 words was very small and they were not counted separately among the number of 5-word NTTs).

C. Construct prior co-word networks for NTTs and notTTs

Most of the words contained in NTTs and notTTs (called "member words") had appeared in some phrases of prior patents in the same IPCs of the focal patent. These phrases were called "prior related terms" of the member words. Then, the member words of NTTs or notTTs and their neighbors (other words that co-occur with the member words in the "prior related terms") were used as nodes, and the co-occurrence relationships in the "prior related terms" were used as edges to construct NTTs or notTTs' previous co-word network (PCN) . Figure 1 shows the construction process of the prior co-word network of NTT "color value table".

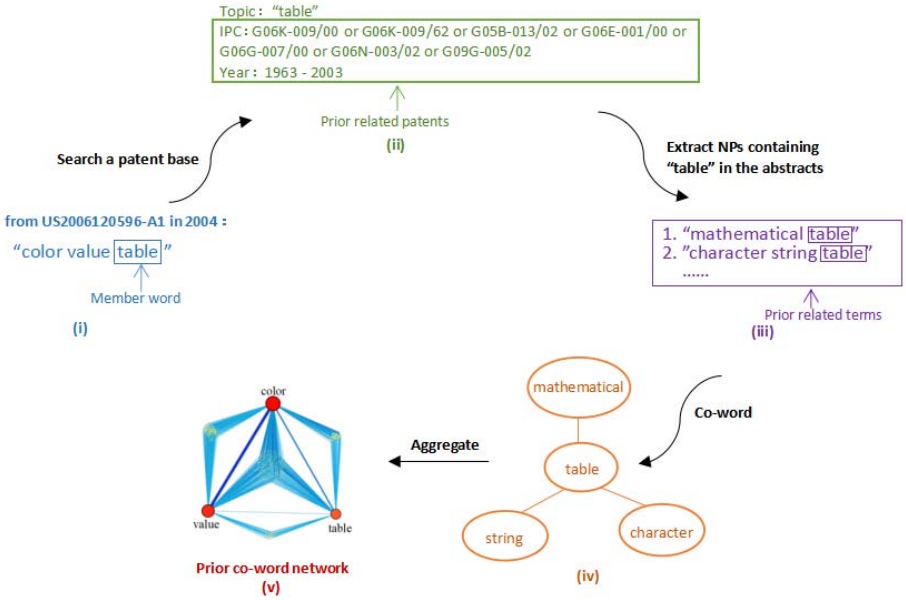

Figure 1. Construction process of sample prior co-word network.

D. Calculate the predictor variables in the PCN

The atypicality and conventionality of member word combinations [10] may affect the formation of NTTs. Therefore, we summarized previous studies and set percentage of edges with zero weight (*zero_proto*) and mean edge weight between member words (*proto_mean*) to represent the atypicality and conventionality of member combinations in conjunction with complex network knowledge. Moreover, Liben-Nowell and Kleinberg [11] demonstrated that two common neighbor nodes in a network were more likely to form between new connections. Based on this, we set the mean common neighbors of member words (*com_mean*) variable according to this view. Finally, the number of hub nodes (*hubs*) variable was set to capture the characteristics of the prior co-word network.

1) zero_proto. If two member words in NTTs or notTTs had never appeared together in "previous related terms", the edge weight between the two member word nodes was considered to be zero. zero_proto calculated the number of member word combinations with zero edge weight as a proportion of the number of all combinations. The calculation formula was as follows:

$$zero\_proto = \frac{x}{C_n^2} \tag{1}$$

where x represented the number of edges with zero weight and n represents the number of member words in NTTs or notTTs.

2) proto_mean. This variable represented the average of the weights of all edges between member words. The calculation formula was as follows:

$$proto\_mean = \left(\sum_{1\le i,j\le n, i\ne j} \alpha_{i,j}\right) / C_n^2 \tag{2}$$

Authorized licensed use limited to: VIT University. Downloaded on June 14,2024 at 06:05:56 UTC from IEEE Xplore. Restrictions apply.

where let $\alpha_{i,j}$ represented the edge weight between w_i and w_j (member words denoted by $w_1, ..., w_n$) in *PCN*, *i, j = 1, ..., n.*

3) com_mean. This variable calculated the mean value of the number of common neighbors for all member word pairs contained in NTTs or notTTs. The calculation formula was as follows:

$$com\_mean = (\sum_{1\le i,j\le n, i\ne j} |\gamma_{i,j}|) / \ C_n^2 \quad (3)$$

where $\gamma_{i,j}$ denoted the common neighbors of w_i and w_j in *PCN, i, j = 1, ..., n.*

4) Hubs. This variable counted the number of hub nodes in the network that were common neighbors with all three member words in an NTTs or notTTs at the same time. Since this variable related to the relationship of 3 member words and was not applicable to two-word NTTs, and the number of 2-word NTTs was small, 2-word NTTs were not considered in this study. The calculation formula was as follows:

$$hubs = \sum_{1\le h,i,j\le n, h\ne i\ne j} |\gamma_{h,i} \cap \gamma_{i,j}| \quad (4)$$

In addition to the above four independent variables, considering that the priority year of the focal patent where the NTTs were located may also affect the formation of NTTs, it was named *year* as a control variable.

E. Build ML-based prediction models for NTTs

Based on the predictor variables constructed in the previous section, we applied these variables to multiple machine learning methods, and selected the machine learning method with the best results to construct a prediction model for NTTs. Specifically, we used classification trees (CTs), random forest (RF), multilayer perceptrons (MLP), and stacking, which were the more popular machine learning methods currently used for classification problems. 80% of the total data was randomly selected as the training set, and the remaining 20% was used as the test set to evaluate the machine learning methods.

CTs, as a basic machine learning classification algorithm, was a common method to solve classification problems. In this study, CART was used to train the classification tree by using the gini index to select the classification features.

RF was an algorithm that integrates multiple CTs through the idea of integration, and we set the number of trees in the random forest to 100.

MLP was a neural network model that contains at least one hidden layer in the middle of the input and output layers. An activation function existed for each node in the hidden layer to process the data. The MLP model we built used two hidden layers, the activation function was set to a softmax function, and cross entropy was used as the error function during the training process.

The stacking algorithm used random forest, AdaBoost, gradient boost classification tree, ExtraTrees, and support vector classifier (SVC) as five base estimators. The classification tree was used as the final estimator to output of the final prediction results. The final estimator was trained using ten-fold cross-validated predictions of the base estimators.

III. RESULTS AND EVALUATION OF MACHINE LEARNING METHODS

The key evaluation metrics of prediction models constructed by machine learning algorithms, CTs, RL, MLP, and stacking, were examined separately. We chose accuracy, precision, recall, F1-score, and AUC as the evaluation metrics. Table 2 shows the results, and it was obvious that among all machine learning models, the prediction models constructed by the stacking algorithm outperformed the other algorithms in all evaluation metrics. It further illustrated that the stacking algorithm, as an integrated algorithm, better integrated the advantages of the machine learning algorithms included in its base estimators and final estimator to obtain better results in the classification problem [12]. Therefore, we chose the stacking algorithm to construct the NTTs prediction model.

Table 1. Confusion matrices of ML-based NTT prediction models.

Predicted	**Observed**							
	Classification trees (CTs)		**Random Forest (RF)**		**Multilayer perceptron (MLP)**		**Stacking method**	
	0	1	0	1	0	1	0	1
0	85	25	80	25	83	33	101	16
1	40	73	34	84	22	80	32	75

Authorized licensed use limited to: VIT University. Downloaded on June 14,2024 at 06:05:56 UTC from IEEE Xplore. Restrictions apply.

Table 2. Key metrics of NTT prediction models.

Output Rating	Test set				
	Accuracy	Precision	Recall	F1-score	AUC
CTs	0.71	0.74	0.65	0.69	0.71
RF	0.74	0.71	0.77	0.74	0.74
MLP	0.75	0.78	0.71	0.74	0.77
Stacking method	0.79	0.78	0.79	0.79	0.79

We conducted validation experiments to examine the feasibility of the prediction model, which used data up to 2015. We first retrieved the patents since 2016 from DII database, and randomly extracted 29 NTTs and notTTs, respectively. After that, the prediction model formed by NTTs were used to predict the 58 phrases. The prediction accuracy rate reached 86.2%, which validated the feasibility of the model.

To facilitate a better understanding of the application of this research model in a practical process, we demonstrate the application of the model in the enterprise, as shown in Figure 2. The model helps R&D staff generate new ideas by analyzing a large amount of historical data from the enterprise and predicting possible future NTTs. Figure 3 represents two approaches to generate new NTTs. One is the traditional method that relies on experience through continuous exchange and cooperation between R&D personnel and NTT prediction models (red route in Figure. 2). The other is an automated method that extracts NTTs from a large amount of technical file data using the prediction model proposed in this study (blue route in Figure. 2).

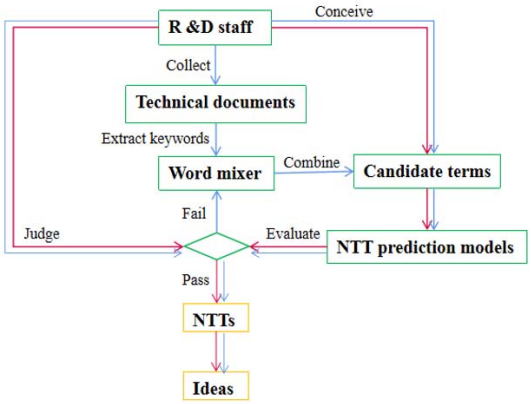

Figure 2. An application of NTTs prediction model.

IV.CONCLUSIONS

NTTs provide a new perspective for unveiling the "fuzzy front end" of innovation process. At the same time, the study of NTTs, as the predecessor of technical terminology, helps to improve the understanding of the development process of technical terminology. In this study, by combining the knowledge of complex networks, we find the variables that may affect the formation of NTTs and compare several machine learning algorithms to build a prediction model of NTTs with a prediction accuracy of 78.6% by selecting the stacking algorithm. The model essentially uses big data to drive idea generation. This helps speed up the R&D process and identify R&D opportunities early so that we can get a head start in the fierce technology competition. However, it is undeniable that this model still has some limitations, firstly, only data from a single domain is used to construct the model for the generalizability of the model needs to be further investigated. Secondly, although the number of 2-word NTTs is small, there is still a need for future research to investigate the formation of 2-word NTTs to expand the applicability of the prediction model and better assist the practical work.

REFERENCES

[1] Fleming, L., "Recombinant uncertainty in technological search," Management Science 47, 117–132 (2001).

[2] Schoenmakers, W. and Duysters, G. "The technological origins of radical inventions," Research Policy 39(8), 1051-1059 (2010).

[3] Damerau, F. J. "Generating and evaluating domain-oriented multi-word terms from texts," Information Processing and Management 29(4), 433–447 (1993).

[4] Dunning, T. "Accurate methods for the statistics of surprise and coincidence," Computational Linguistics 19(1), 61–74 (1993).

[5] Frantzi, K., Ananiadou, S. and Mima, H. "Automatic recognition of multi-word terms: The C-value/NC-value method," International Journal of Digital Libraries 3(2), 117–132 (2000).

[6] Vàzquez, M. and Oliver, A. "Improving term candidates selection using terminological tokens," Terminology 24(1), 122-47 (2018).

[7] Liu, J., Shang, J, Wang C., Ren, X. and Han, JW. "Mining quality phrases from massive text corpora," Proc. SIGMOD, 1729–1744 (2015).

[8] Ünal, H. T., & Başçiftçi, F. "Evolutionary design of neural network architectures: a review of three decades of research," Artificial Intelligence Review, 55, 1723–1802 (2022).

[9] Wang, X., Qiu, P., Zhu, D., Mitkova, L., Lei, M. and Porter, A. L. "Identification of technology development trends based on subject–action–object analysis: The case of dye-sensitized solar cells," Technological Forecasting and Social Change 98, 24–46 (2015)..

[10] Uzzi, B., Mukherjee, S., Stringer, M., & Jones, B. (2013). Atypical combinations and scientific impact. Science, 342(6157), 468-472.

[11] Liben-Nowell, D. and Kleinberg, J. "The link-prediction problem for social networks," Journal of the American Society for Information Science and Technology, 58(7), 1019–1031 (2007).

[12] Mohammed, A. S. , Asteris, P. G. , Koopialipoor, M. , Alexakis, D. E. and Armaghani, D. J. "Stacking ensemble tree models to predict energy performance in residential buildings," Sustainability, 13(15) (2021).

Authorized licensed use limited to: VIT University. Downloaded on June 14,2024 at 06:05:56 UTC from IEEE Xplore. Restrictions apply.