An Approach Merging the IDM-Related Knowledge

Xin Ni[1](✉), Ahmed Samet[2](✉), and Denis Cavallucci[1](✉)

[1] ICUBE/CSIP, INSA of Strasbourg, 24 Boulevard de la Victoire,
67084 Strasbourg, France
{xin.ni,denis.cavallucci}@insa-strasbourg.fr

[2] ICUBE/SDC, INSA of Strasbourg, 300 Bd Sébastien Brant,
67412 Illkirch, France
ahmed.samet@insa-strasbourg.fr

Abstract. Patents are one of the main innovation knowledge sources for engineers and companies. Inventive Design Method (IDM) – results from a research that extends from TRIZ and contains formal knowledge description components using ontologies, such as problems, partial solutions, and parameters. In this paper, we introduce IDM-Similar model that extends existing research work in IDM-related knowledge. A neural network named Word2vec and cosine similarity approach are used to build this model to compute the similarity among problems in wide range domains' patents covering from the chemistry to mechanics and the computer to physics. Our model assumes that a partial solution of a patent could be used to solve the problem of another patent from a different domain if these two problems are similar enough. Experiments show that our model is a promising alternative to classical TRIZ for engineers to associate their problems in a field to solutions from patents of another field. Consequently, the step dedicated to solution concepts ideation is improved using our work.

Keywords: TRIZ · Inventive Design Method · Word2vec · Similarity computation

1 Introduction

TRIZ is the theory of inventive problem solving [3] and its derived approach Inventive Design Method (IDM) was created to assist engineers in their invention process [2]. Moreover, IDM-related knowledge mainly is hidden in patent documents. And also, more than 80% of man's technical knowledge is described in patent literature [2] and the World Intellectual Property Organization revealed that 90% to 95% of all the world's inventions are found in patent documents [1]. Therefore, the patents now are the most useful reference materials for engineers to find out innovative solutions.

However, most companies, nowadays, still rely on engineers' experience or brainstorming among different domain's experts or manual work on searching knowledge from patent documents to promote the advancement of products. These types of methods now cannot fit the current rise of infinite and permanent renewal of

Published by Springer Nature Switzerland AG 2019
R. Benmoussa et al. (Eds.): TFC 2019, IFIP AICT 572, pp. 147–158, 2019.
https://doi.org/10.1007/978-3-030-32497-1_13

information and data's throughout all domains. At the same time, engineers are traditionally using the patent to check if their idea has already been exploited by somebody else or to browse large quantity of images to check the state of the art around their project in order to be aware of what exists. Further, some innovative solutions or methods might be from other domains, which normally could not be found by engineers via conventional search methods because their lack of expertise. The limitation of the knowledge that engineers have and the inefficiency of searching patent documents have become a big obstacle to promote the development of innovation.

Thus, finding an innovative solution for some special industrial problem from a wide range of patent documents is of paramount importance. Moreover, engineers usually are unable to find out breakthrough inventive solutions, key to competitiveness, using classical methods. To address this issue, we propose to use Google's neural network named word2vec [6] to obtain the sentence vector for every problem in patents. Then, we compute the cosine similarity between the pairs of sentences in order to retrieve similar problems from other patent documents. Consequently, we can successfully merge three of the similar IDM-related knowledge from patent documents that are problems, partial solutions, and parameters [4]. This approach should be helpful to engineers to easily and effectively find out similar problems from other patent documents. The latter could suggest more inventive solutions mined from a wider range of industrial domains so that they could be better assisted in Research & Design development activities [3, 5].

The final experimental results on real-world dataset show that the approach we used achieve promising results on finding similar problems from a large amount of patent documents as well as corresponding solutions. This greatly speeds up the whole efficiency for seeking innovative solutions and emerging IDM-related knowledge. Particularly, we show that our work successfully finds out two typical similar problems that are just matched with the vibration and data retention problems as well as corresponding solutions from different domain's patents. The contribution of this paper is that we mix the use of neural networks and IDM-knowledge to facilitate the merging of IDM-knowledge of different domains' patent documents in order to promote R&D activities. Besides, no other work, to the best of our knowledge, has been introduced so far.

The paper consists of the following sections. Section 2 introduces a brief state of art about similarity computation on patent documents. Section 3 and Sect. 4 separately details the IDM ontology as well as the methodology of the Word2vec, Sentence2vec, and similarity computation. We expose the experiments validating our model and the case study detailly in Sect. 5. We finally conclude our work and show perspectives for future works.

2 Related Work

Similarity computation is one of the most important tasks in natural language process. At the same time, there are many research achievements that use content of patent documents. These research achievements usually are valuable for product innovation. Significant efforts have recently been recorded in the similarity computation on patent documents, especially in word similarity computation [7, 8, 10] and document similarity computation [11–18].

In the study of word similarity computation, Dagan et al. [7], at 1993 year, proposed that the feature according to contextual information in documents can be used to better compute the similarity among words when we have a large-scale corpus. Most of the following research is then based on this achievement to further improve the final performance. Further, Pilehvar et al. [8] used an iterative method for calculating topic-sensitive PageRank [9] to construct each semantic signature in order to compute the similarity between two words. In addition, Terra et al. [10] investigated frequency estimates for co-occurrence that are based both on documents and on a variety of different window sizes, and examine the impact of the corpus size on the frequency estimates. They found that the size of a context for the target word will notably affect the result of the similarity computation on words.

The study of the document similarity computation is mainly as follows. In the beginning, Kessler et al. [11] and Small et al. [12] separately proposed Bibliographic Coupling and Co-citation Analysis methods at 1963 and 1973 year to analyze the similarity among different patents. Besides, a patent classification system using co-citation analysis has been also proposed by Lai et al. [13] to compute the patent similarity. Further, McGill et al. [14] and Mowery et al. [15] computed the similarity of firm patents via cross-citation rate when analyzing patent citation data. Moehrle et al [16] and Bergmann et al [17] also used natural language process methods to extract a subject–action–object–format (SAO) structures in patents first and then built similarity matrices for patents to evaluate their similarity. In addition, some indexes as centrality index, technology cycle index, and technology keyword clusters in patents are also used for in-depth quantitative analysis in order to compute the patent similarity [18].

The above similarity computation approaches on patent documents now mainly are applied on wide fields like evaluating the risk of patent infringement [17], discovering competitive intelligence [19], identifying technology opportunities [20], measuring the novelty of patents [21], making the technological roadmap [22], detecting the similarity between patent documents, and scientific publications [23], etc. Our work achieves also some inspiration from the above similarity computation methods that are used on real applications. However, we found that few of existing similarity computation methods or models which have been used on IDM-related knowledge, especially computing the similarity among different problems in patents. In this paper, we try to fill the gap of similarity computation on the field of IDM-related knowledge implementation by using the word vector model in order to improve the efficiency of finding similar problems and corresponding solutions from patent documents as well as expand the border of finding solutions.

3 IDM Ontology

Inventive Design Method (IDM) is based on the Theory of Inventive Problem Solving - the translation of the acronym TRIZ. It represents an extension of TRIZ and is usually perceived as more guided, therefore easier to teach to others since it is more formally described. It aims at assisting companies and engineers to solve complex and multi-disciplinary problems in creative ways.

Different from other ontologies, IDM ontology is generic and applicable in all fields [25]. In addition, Cavallucci et al. [26] proposed the main concepts of IDM that are problems, partial solutions, and contradictions including element parameters and values. In patents, problems normally describe unsatisfactory features of existing methods or situations. Partial solutions provide improvements or changes to the defined problems. Each problem may cause one or more contradictions the patent solves. Besides, partial solutions must be the simplest possible. Elements are components of the system and parameters qualify the element with certain specifics. Parameters are also qualified by values. The knowledge contained in these key concepts of patents usually has great value for engineers. Thus, we try to maximize this knowledge, hidden in patents unstructured text, to find out relevant inventive and potential solutions to a given problem in order to assist creative R&D departments.

4 Methodology

In this section, we introduce a new model named IDM-Similar for IDM-related knowledge similarity computation. Our work aims to find out similar problems to a given problem from the large-scale patent corpus in order to merge IDM-related knowledge. We first extract problems as well as corresponding partial solutions and parameters from several patent corpus using IDM-related knowledge extraction tool [24]. Then, we compute the similarity between these problems.

Formally, the extracted IDM-related knowledge set $K_i = \{P_i, PS_i, PA_i\}$ is from i-th patent document where P_i, PS_i, and PA_i are problems, partial solutions, and parameters respectively in the i-th patent document. Given the j-th problem $P_i^j = \left(w_i^{j_1}, w_i^{j_2}, \ldots, w_i^{j_{|w_i|}}\right)$ in the i-th patent document where $w_i^{j_{|w_i|}}$ is the $|w_i|$-th word in the j-th problem sentence, and we compute its similarity between to the other considered problems P.

4.1 Word Vector

We aim at using Word2vec to compute every word's vector in training corpus. Indeed, we integrate open source Wikipedia corpus to train Word2vec model. Word2vec is a type of word embedding model that proposed by Google first. This two-layer neural network can be trained by a large-scale corpus to achieve a vector in the space for each unique word in the corpus. Word vector is positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space [6]. The trained Word2vec model can simplify the processing of the considered text into n-dimensional space vector operation. Thereby, the similarity in vector space can represent the semantic similarity of text. In addition, the process of training on Word2vec is unsupervised and this two-layer neural network turns text into a numerical form that deep networks can understand so that it can be run on the computer efficiently.

The continuous bag-of-words (CBOW) and skip-gram [6] are two types of model architectures that are used on Word2vec to produce a distributed representation of words. These two architectures are similar at an algorithmic-wise but the main difference between them is that CBOW predicts the target word according to the context words around the original word. On the contrary, skip-gram predicts each context word via the target word. For instance, as shown in Fig. 1, CBOW predicts that the blank (target word) is "*mat*" when it encounters "*The dog sits on the _. It looks like a cat.*". For skip-gram, it will predict that the context of the center word "*mat*" is "*The dog sits on the*" and "*It looks like a cat*" when it reaches "_ _ _ _ _ *mat*.-_ _ _ _ _". Further, the CBOW smoothly processes the distributed information, such as treating a whole piece of context as a single observation set. In many cases, it is normally helpful for processing the small-sized dataset. In contrast, the skip-gram combines every context and target word as a new observation set and it normally works well on the large-sized dataset. Moreover, skip-gram does a better job for infrequent words. In this paper, we use Word2vec that is with skip-gram because there are many infrequent words in Wikipedia corpus comparatively to any other corpus.

4.2 Sentence Vector

As illustrated in Fig. 2, we choose to work with unsupervised text matching method to obtain each sentence's vector. Matched sentence problems in patent documents are extracted by the IDM-related knowledge extraction tool. We achieve the sentence vector $\vec{\boldsymbol{P}}$ by calculating the average vector of all words in each sentence. The calculation is defined as:

$$\vec{\boldsymbol{P}} = \frac{\sum_{i=0}^{j} \overrightarrow{\boldsymbol{w}_{\iota}}}{j} \tag{1}$$

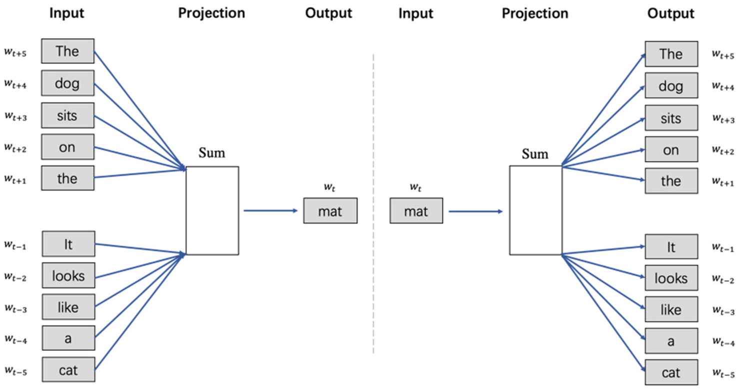


Fig. 1. CBOW (left) and skip-gram (right)

4.3 Cosine Similarity

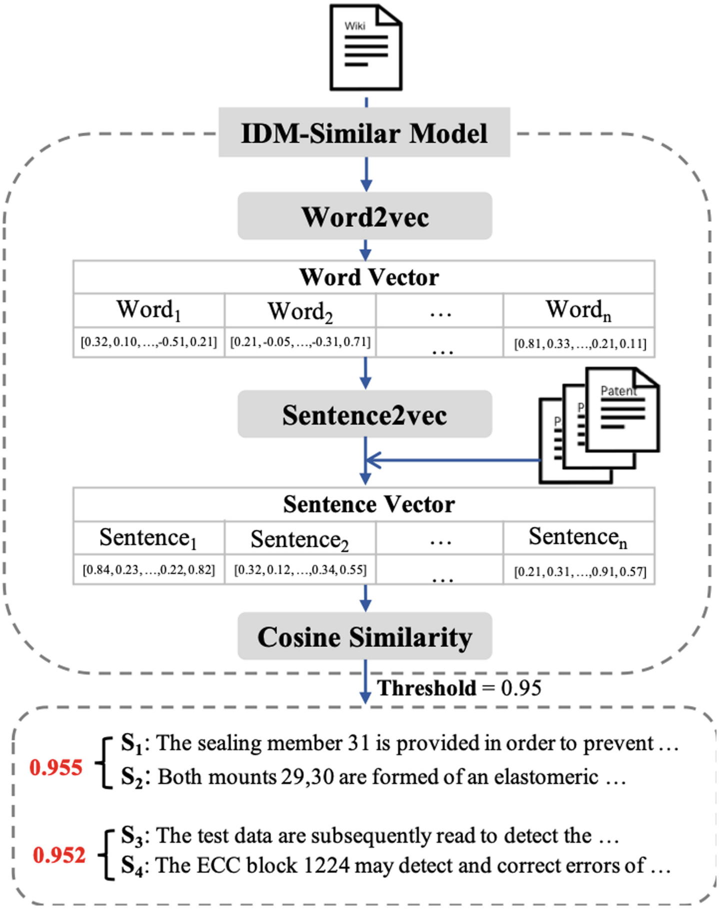
We first compute the cosine distance between the given problem's sentence vector $\overrightarrow{\boldsymbol{P_i}}$ and another sentence vector $\overrightarrow{\boldsymbol{P_j}}$:

$$CosineDistance = \frac{\overrightarrow{\boldsymbol{P_i}} \cdot \overrightarrow{\boldsymbol{P_j}}}{|\overrightarrow{\boldsymbol{P_i}}||\overrightarrow{\boldsymbol{P_j}}|} = \frac{\sum_{i,j=1}^{n} \boldsymbol{P}_i \times \boldsymbol{P}_j}{\sqrt{\sum_{i=1}^{n} (\boldsymbol{P}_i)^2} \times \sqrt{\sum_{j=1}^{n} (\boldsymbol{P}_j)^2}} \tag{2}$$

Next, the cosine similarity is defined as:

$$Cosine\ Similarity = 1 - Cosine\ Distance = 1 - \frac{\overrightarrow{\boldsymbol{P_i}} \cdot \overrightarrow{\boldsymbol{P_j}}}{|\overrightarrow{\boldsymbol{P_i}}||\overrightarrow{\boldsymbol{P_j}}|} \tag{3}$$

Overall, if the value of cosine similarity is closer to 1, the similarity between pairs of sentences increases.



Word Vector			
Word_1	Word_2	…	Word_n
[0.32, 0.10, …, -0.51, 0.21]	[0.21, -0.05, …, -0.31, 0.71]	…	[0.81, 0.33, …, 0.21, 0.11]

Sentence Vector			
Sentence_1	Sentence_2	…	Sentence_n
[0.84, 0.23, …, 0.22, 0.82]	[0.32, 0.12, …, 0.34, 0.55]	…	[0.21, 0.31, …, 0.91, 0.57]

0.955 { $\mathbf{S_1}$: The sealing member 31 is provided in order to prevent …
$\mathbf{S_2}$: Both mounts 29,30 are formed of an elastomeric …

0.952 { $\mathbf{S_3}$: The test data are subsequently read to detect the …
$\mathbf{S_4}$: The ECC block 1224 may detect and correct errors of …

Fig. 2. An overview of our model

As illustrated in Fig. 2, we first employ the open source Wikipedia corpus to train Google's neural network Word2vec [6] in order to achieve every word's vector. After that, we adopt the trained word vector model to obtain the sentence vector for every inputting problem sentence. Finally, we apply the similarity computation approach to calculate the cosine similarity among each pairs of problem sentences and then find out those similar problems whose value are greater than the threshold as well as their corresponding solutions and parameters.

5 Experiments

5.1 Datasets and Evaluation Metrics

In this paper, we use the clean version of the English Wikipedia dataset [27] to train our Word2vec model. It only contains regular article text but removes tables and links to foreign language versions. Also, in this dataset, citations, footnotes, and markup were removed as well as hypertext links were converted to ordinary text. Furthermore, we also evaluate the similarity computation method on the Utility Patent datasets of US Patent Grant dataset [28].

US patents are mainly classified to three types: utility patent, design patent, and plant patent. According to the USPTO, utility patents are granted to anyone who invents or discovers any new and useful process, machine, article of manufacture, or composition of matter, or any new and useful improvement. Design patents are granted to anyone who invents a new, original, and ornamental design for an article of manufacture. Plant patents are granted to anyone who invents or discovers and asexually reproduces any distinct and new variety of plant. But 90% of US patents are utility patents, which protect the utility or functional aspects of an invention and they normally contain some kind of similarity on invention domains with each other compared to other types of patents. Therefore, in this paper, we use utility patent dataset as the test dataset to check the performance of the similarity computation approach. Utility patent dataset contains a total of 6,161 documents.

Finding the "gold-standard" ground truth of verifying the similarity among different sentences always is an open problem, especially for the sentences that are from different domains, and no work solved this problem. In this paper, we referred to 3 experts who are respectively from mechanics, chemistry, and architecture to verify the experimental results manually. A cross-checking among them is made to ensure the authenticity of the final results.

5.2 Experimental Settings

In order to achieve an optimal word vector model, in this paper, we tune our Word2vec model on the training dataset by using different parameters. At the same time, we also optimize the efficiency and accuracy for training the model. The final parameters of our Word2vec we set are shown in Table 1.

The number of dimensions in the created vectors is defined by #*size*. Hence size here means each document receives a 100-dimensional vector from training.

More dimensions usually mean slower training and we may risk having overfitting when the model works on small-sized datasets. The #*window* indicates how many words before and after a given word would be included as context words of the given word. Those words in training corpus that the minimum frequency of words is below the threshold of frequency will be discarded by #*min_count*. It is helpful to filter out those extremely rare or wrong words in the corpus. Word2vec model will use the hierarchical softmax as the loss function when we set #*negative* > 0 and #*hs* = 1 as well as we set 3 noise words in the model when the value of #*negative* is 3. The #*sample* represents the threshold of sampling. Those words with higher frequency in the training dataset will be randomly down-sampled.

Furthermore, in the cosine similarity computation, as mentioned in Sect. 4, if the similarity value between the two sentences is closer to 1, they are considered as similar. We finally fix the threshold of the similarity to 0.95 for performance reasons upon carrying out several tests and optimizing the size of the output.

Table 1. Parameters of the Word2vec model

Parameter	#size	#window	#min_count	#negative	#sample	#hs
Value	100	5	5	3	0.001	1

5.3 Overall Results

In this part, we carry out the performance analysis of our model on the US patent dataset and offer some further analysis. At first, as illustrated in Table 2, Patent Extractor [24] we used has extracted three types of IDM-related knowledge from 6,161 US patent documents. We used 4,574 problems among them as input dataset to Sentence2vec model. We compute the similarity between any two different problems via IDM-Similar model. The performance of our model on US patent dataset is shown in Table 3. From the results, we can observe that IDM-Similar model finally gives us 1,121 pairs of similar problems when the similarity threshold is set as 0.95. Through three experts' cross-checking, the number of true positive (TP) and false positive (FP) of final results are 1,000 and 121 respectively so that the precision of similarity is 89.21%. It demonstrates that our model can efficiently find out similar IDM-related knowledge from a large amount of patent documents belonging to different domains.

Table 2. Performance of Patent Extractor on US patent dataset

Model	Patent Extractor		
IDM-related knowledge	Problem	Partial solution	Parameter
Number	**4,574**	17,971	29,264

Table 3. Experimental results on US patent dataset

Model	IDM-Similar			
Metric	TP	FP	Total	Precision
Number	1,000	121	1,121	**89.21%**

5.4 Case Study

The objective of this part is to demonstrate the practical value of our model on merging IDM-related knowledge from different domains' patents. Two case studies on chemistry/mechanics and computer/physics domains respectively assess the performance of our model.

1. **Chemistry/Mechanics:** "Collector for bipolar lithium ion secondary batteries (US9537152B2)" and "Vacuum cleaner with motor between separation stages (US9532691B2)" are two US patents that are from chemistry and mechanics respectively. As shown in Fig. 3, our model finds out a pair of similar problems: "*The sealing member 31 is provided in order to* ***prevent contact*** *between the current collectors 11 adjacent to each other inside the battery and prevent a short circuit caused by slight unevenness at edge portions of the single cell layers 19 in the power generation element 21.*" and "*Both mounts 29, 30 are formed of an elastomeric material and act to* ***isolate*** *the second dirt-separation stage 7 and thus the remainder of the dirt separator 3 from the vibration generated by the vacuum motor 6.*" After analyzing the whole two patents, experts think that both problems are linked and belong to different domains. It is possible to solve the short circuit problem in US9537152B2 patent with the provided solution of the elastomeric material in US9532691B2 patent and vice versa.

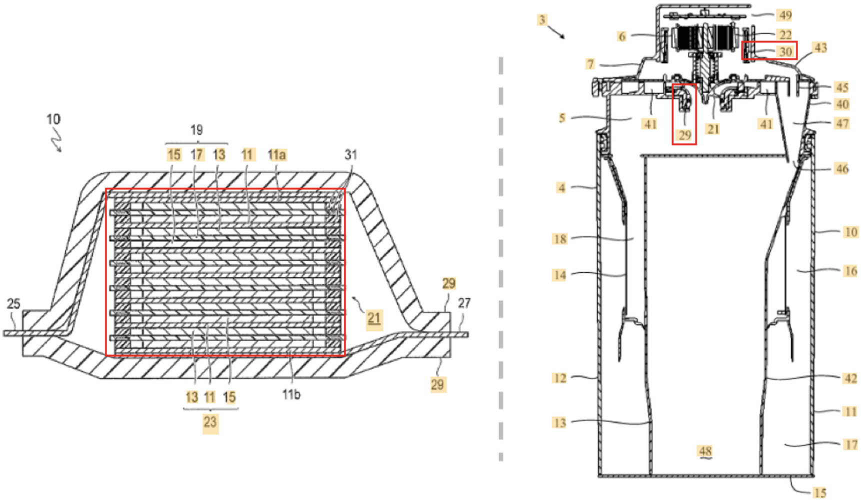


Fig. 3. Diagrams of the sealing member (left) and the elastomeric material (right)

2. **Computer/Physics:** “Hybrid-HDD with improved data retention (US9536619B2)” and “Semiconductor device and method of fabricating the same (US9536897B2)” are two US patents that are from computer and physics respectively. Two similar problems our model found are illustrated in Fig. 4: “*The test data are subsequently read to detect the possibility of data retention* ***errors*** *that may occur when reading the associated* ***user data****.*” and “*The ECC block 1224 may* ***detect and correct errors of data*** *which are read out from the memory device 1210.*” We think there is a kind of possibility to add ECC block using US9536897B2 patent into the left device to solve the data retention errors that mentioned in US9536619B2 patent. This case is from two similar domains and also was extracted by our model successfully.

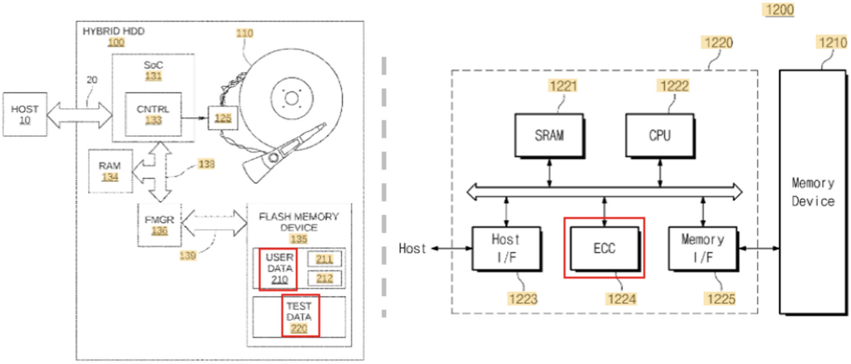


Fig. 4. Diagrams of the hybrid HDD (left) and the memory systems (right)

In conclusion of these cases, we note that the final similar problems our IDM-Similar model found from different domains’ patents have a significant practical value for inventive solutions. In fact, these experimental results on the real-world dataset are a great encouragement for us.

6 Conclusion and Future Work

In this paper, we propose an IDM-Similar model to compute the similarity of the different pairs of sentences in order to merge the IDM-related knowledge that are from different domains’ patent documents. Our model can make full use of the IDM-related knowledge in patents and find out similar problems from different domains. In the experimental results, we show that our model has good precision and significant practical value. In particular, we demonstrate two real cases that the problems can be solved by inventive solutions from another domain using our model. It will significantly improve the efficiency for engineers to find out innovative solutions and promote R&D activities for companies.

In the future, we will explore the following directions:

1. An accurate IDM-related knowledge extractor can further improve the performance of our model. We will explore how to effectively and accurately extract IDM-related knowledge from unstructured patent documents to further enhance the performance.
2. The different size and types of training datasets can affect the performance of the Word2vec model. We will try to utilize some larger patent corpus to train Word2vec model to improve its performance.

Acknowledgments. This work is supported by China Scholarship Council (CSC).

References

1. Yeap, T., Loo, G.H., Pang, S.: Computational patent mapping: Intelligent agents for nanotechnology. In: International Conference on Mems, Nano and Smart Systems. IEEE (2003)
2. Souili, A., Cavallucci, D., Rousselot, F.: A lexico-syntactic pattern matching method to extract IDM-TRIZ knowledge from on-line patent databases. Procedia Eng. **131**(Complete), 418–425 (2015)
3. Altshuller, G.S.: 40 Principles: TRIZ keys to technical innovation (Lev Shulyak et Steven Rodman, Trans.), 1st edn, 141p. Technical Innovation Center, Inc., Worcester (1998). ISBN-10: 0964074036
4. Zanni-Merk, C., Cavallucci, D., Rousselot, F.: Using patents to populate an inventive design ontology, Proceedings of the TRIZ Future Conference 2010, Bergamo, 3–5 November 2010, pp. 52-62. Elsevier Ltd. (2010)
5. Cavallucci, D., Khomenko, N.: From TRIZ to OTSM-TRIZ, Addressing complexity challenges in Inventive design. Int. J. Prod. Dev. **4**, 4–21 (2007)
6. Mikolov, T., et al.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
7. Dagan, I., Marcus, S., Markovitch, S.: Contextual word similarity and estimation from sparse data. In: Proceedings of the 31st annual meeting on Association for Computational Linguistics. Association for Computational Linguistics (1993)
8. Pilehvar, M.T., Jurgens, D., Navigli, R.: Align, disambiguate and walk: a unified approach for measuring semantic similarity. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1 (2013)
9. Haveliwala, T.H.: Topic-sensitive PageRank. In: Proceedings of the 11th international conference on World Wide Web. ACM (2002)
10. Terra, E., Clarke, C.L.A.: Frequency estimates for statistical word similarity measures. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. Association for Computational Linguistics (2003)
11. Kessler, M.M.: An experimental study of bibliographic coupling between technical papers. No. 62 673TN1. Massachusetts Inst of Tech Lexington Lincoln Lab (1962)
12. Small, H.: Co-citation in the scientific literature: a new measure of the relationship between two documents. J. Am. Soc. Inf. Sci. **24**(4), 265–269 (1973)
13. Lai, K.-K., Shiao-Jun, W.: Using the patent co-citation approach to establish a new patent classification system. Inf. Process. Manag. **41**(2), 313–330 (2005)

14. McGill, J.P.: Technological knowledge and governance in alliances among competitors. Int. J. Technol. Manag. **38**(1), 69 (2007)
15. Mowery, D.C., Oxley, J.E., Silverman, B.S.: Technological overlap and interfirm cooperation: implications for the resource-based view of the firm. Res. Policy **27**(5), 507–523 (1998)
16. Moehrle, M.G., et al.: Patent-based inventor profiles as a basis for human resource decisions in research and development. R&D Manag. **35**(5), 513–524 (2005)
17. Bergmann, I., et al.: Evaluating the risk of patent infringement by means of semantic patent analysis: the case of DNA chips. R&D Manag. **38**(5), 550–562 (2008)
18. Yoon, B., Park, Y.: A text-mining-based patent network: analytical tool for high-technology trend. J. High Technol. Manag. Res. **15**(1), 37–50 (2004)
19. Shih, M.-J., Liu, D.-R., Hsu, M.-L.: Discovering competitive intelligence by mining changes in patent trends. Expert Syst. Appl. **37**(4), 2882–2890 (2010)
20. Yoon, B., Park, Y.: A systematic approach for identifying technology opportunities: keyword-based morphology analysis. Technol. Forecast. Soc. Change **72**(2), 145–160 (2005)
21. Gerken, J.M., Moehrle, M.G.: A new instrument for technology monitoring: novelty in patents measured by semantic patent analysis. Scientometrics **91**(3), 645–670 (2012)
22. Lee, S., et al.: Business planning based on technological capabilities: patent analysis for technology-driven roadmapping. Technol. Forecast. Soc. Chang. **76**(6), 769–786 (2009)
23. Magerman, T., Van Looy, B., Song, X.: Exploring the feasibility and accuracy of Latent Semantic Analysis based text mining techniques to detect similarity between patent documents and scientific publications. Scientometrics **82**(2), 289–306 (2009)
24. Souili, A., Cavallucci, D.: Automated extraction of knowledge useful to populate inventive design ontology from patents. TRIZ – The Theory of Inventive Problem Solving, pp. 43–62. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56593-4_2
25. Bultey, A., De Bertrand De Beuvron, F., Rousselot, F.: A substance-field ontology to support the TRIZ thinking approach. Int. J. Comput. Appl. Technol. **30**(1), 113–124 (2007)
26. Cavallucci, D., Rousselot, F., Zanni, C.: Initial situation analysis through problem graph. CIRP J. Manuf. Sci. Technol. **2**(4), 310–317 (2010)
27. Wikipedia Dataset. http://mattmahoney.net/dc/textdata.html. Accessed 13 Apr 2019
28. US Patent Dataset. https://bulkdata.uspto.gov/data/patent/grant/redbook/fulltext/2017/. Accessed 13 Apr 2019