J Inf Process Syst, Vol.19, No.5, pp.631~640, October 2023
https://doi.org/10.3745/JIPS.02.0205

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Research on Keyword-Overlap Similarity Algorithm Optimization in Short English Text Based on Lexical Chunk Theory

Na Li[1], Cheng Li[2,*], and Honglie Zhang[2]

Abstract

Short-text similarity calculation is one of the hot issues in natural language processing research. The conventional keyword-overlap similarity algorithms merely consider the lexical item information and neglect the effect of the word order. And some of its optimized algorithms combine the word order, but the weights are hard to be determined. In the paper, viewing the keyword-overlap similarity algorithm, the short English text similarity algorithm based on lexical chunk theory (LC-SETSA) is proposed, which introduces the lexical chunk theory existing in cognitive psychology category into the short English text similarity calculation for the first time. The lexical chunks are applied to segment short English texts, and the segmentation results demonstrate the semantic connotation and the fixed word order of the lexical chunks, and then the overlap similarity of the lexical chunks is calculated accordingly. Finally, the comparative experiments are carried out, and the experimental results prove that the proposed algorithm of the paper is feasible, stable, and effective to a large extent.

Keywords

Keyword Overlap, Lexical Chunk Theory, Short English Text, Similarity Algorithm

1. Introduction

The text similarity algorithm has been the research focus for a large amount of overseas and domestic scholars. As early as the 1970s, the algorithm has been used for the information retrieval system for the similarity calculation between the user's retrieval request string and the database text [1]. Through many years of exploration, so far, the text similarity algorithm has been widely applied in the information retrieval, the image retrieval, the automatic text abstract generation as well as the text replication detection and the like [2-4].

With the popularization of the Internet and the quickening pace of people's lives, more and more short texts have emerged in people's horizons. Accordingly, the conventional text similarity calculation algorithms perform an unproductive effect in processing the short text. It is imperative to improve the similarity algorithm and the similarity calculation effect, as it has become a key technology in the natural language processing research.

※ This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Manuscript received December 14, 2022; first revision February 13, 2023; accepted February 26, 2023.

* **Corresponding Author:** Cheng Li (licheng@qqhru.edu.cn)
[1] Public Foreign Language Teaching and Research Department, Qiqihar University, Qiqihar, China (hhln2000@163.com)
[2] College of Computer and Control Engineering, Qiqihar University, Qiqihar, China (licheng@qqhru.edu.cn, 34985155@qq.com)

Copyright© 2023 KIPS

In order to improve the accuracy of short-text similarity calculation, a sea of researchers have adopted the method of expanding the short text information. In [5], the conceptual networks were used to expand the text information for short-text classification. In [6], the dynamic variables were used to obtain the internal relationship to calculate the similarity of short texts. In [7], the similarity of short texts was calculated by constructing a concept tree. The idea of Yin et al. [8] was to construct a model for each cluster, calculate the model parameters through the data object distribution, and employ an appropriate model to measure the similarity of short texts. In [9], the text input was converted into cloze questions containing some form of task description, the pre-trained language models were used to process the questions, and the predicted words were mapped to the tags. In all the above methods, the accuracy of the short-text similarity calculation was improved at the expense of efficiency.

There exist a fleet of research methods of combining the semantic information to improve the accuracy and efficiency of short text similarity calculation. Sun et al. [10] proposed a short-text similarity algorithm based on semantics and maximum matching. Nguyen et al. [11] introduced a new method based on the interdependent representation of short texts to determine their semantic similarity. For this method, each short-text was represented as two dense vectors: the former was constructed using word-to-word similarity based on pre-trained word vectors, and the latter was constructed using word-to-word similarity based on external knowledge sources. Majumder et al. [12] focused on establishing an interpretable semantic text similarity (iSTS) method for a pair of sentences. These methods take the efficiency and accuracy of short-text similarity calculation into account but display a strong dependency on semantic dictionaries.

In recent years, some researchers have introduced the method of machine learning to improve the accuracy of short-text similarity calculation. Liu et al. [13] applied recurrent neural network (RNN) to model the long-term dependence of the input text. Huang et al. [14] proposed a hierarchical convolutional neural network long short-term memory (CNN-LSTM) architecture for sentence modeling, in which CNN was used as an encoder to encode sentences and LSTM was used as a decoder. Peinelt et al. [15] proposed tBERT method to measure the semantic similarity. These methods have achieved good results in similarity accuracy, but the large sample or the long text is required to train the favorable learning model.

The similarity algorithm based on keyword-overlap [16] is an efficient, simple, and easy-to-implement method. For this algorithm, only the word term factor is considered, the important impact of word meaning and word order on the similarity of short texts is ignored, and accordingly the accuracy of all calculations is not high. In this paper, taking the impact of word meaning and word order on the similarity of short texts into consideration, the lexical chunk theory is introduced into the short-text similarity algorithm for the first time, and the short English text similarity algorithm based on lexical chunk theory (LC-SETSA) is put forward, combining the semantic connotation and the fixed word order of lexical chunks. Without the expense of efficiency, the accuracy of short-text similarity calculation is improved from the perspective of semantics and simple word order restriction. In this research, the lexical chunk theory is introduced to optimize the similarity algorithm based on keyword-overlap, and a lexical chunk-based similarity algorithm is proposed to improve the rationality and calculation accuracy of similarity algorithm based on word segmentation, avoiding its efficiency advantage loss. The algorithm proposed of the paper is beneficial to the further optimization of other similarity algorithms based on word segmentation, which is the core significance of our design of the LC-SETSA algorithm. Finally, the experimental results have proved that the proposed LC-SETSA algorithm is feasible, stable, and effective.

The remainder of the paper is organized as follows. The keyword-overlap similarity algorithm is reviewed, and the algorithm defect is analyzed in Section 2. Then, the similarity algorithm based on lexical chunk theory (LC-SETSA) is presented and designed in Section 3. In Section 4, the experimental results and analysis on contrastive data from both the algorithm of the paper and the keyword-overlap similarity algorithm are demonstrated. Finally, conclusions are given with the importance and the practical value of the optimized algorithm as well as its future research directions.

2. Materials and Methods

2.1 Existing Methods

2.1.1 Keyword-overlap algorithm description

Based on the keyword-overlap similarity algorithm, the short text is counted as a collection of independent keywords, and the similarity of two short-texts is estimated by the number of their co-occurrence words. The more co-occurrence words the two short-texts have, the more similar they are. On the contrary, the similarity of the two texts is lower. At the same time, the relative similarity of the two short-texts is guaranteed. The similarity calculation formula is as follows:

$$Sim_a(T_1, T_2) = \frac{2 \times samewords(T_1, T_2)}{Alone(T_1) + Alone(T_2)}, \tag{1}$$

where, $samewords(T_1, T_2)$ is the number of keywords existing in both T_1 and T_2, $Alone(T_1)$ is the number of keywords existing in T_1, and $Alone(T_2)$ is the number of keywords existing in T_2.

2.1.2 Defect analysis of the algorithm

First of all, an example is given to analyze the defects of the keyword-overlap similarity algorithm.

Example 2.1. T_1: *A young girl exclaimed at the policeman nearby.*
T_2: *A young policeman exclaimed at the girl nearby.*

According to Example 2.1, when the algorithm is used to calculate the similarity of T_1 and T_2, there is $Sim_a(T_1, T_2) = 1$ for $samewords(T_1, T_2) = 6$, $Alone(T_1) = 6$ and $Alone(T_2) = 6$.

However, it is illustrated from the Example 2.1 that the semantic expressions of T_1 and T_2 are different, and the reason is that the keyword-overlap similarity algorithm just considers the item information (the keywords of the two sentences are the same), without considering the word order (the keywords are the same sequence for the two sentences). A multitude of research have been done by many scholars on the word order in order to make up for the defects of the keyword-overlap similarity algorithm, and the similarity algorithm is put forward by combining the word item with the word order.

$$Sim_b(T_1, T_2) = \alpha \times Sim_{cp}(T_1, T_2) + (1 - \alpha) \times Sim_{cx}(T_1, T_2) \tag{2}$$

where, $Sim_{cp}(T_1, T_2)$ is the word item similarity value, $Sim_{cx}(T_1, T_2)$ is the word order similarity value,

and α is the weight. The defects of the keyword-overlap similarity algorithm are overcome preferably. However, it is difficult to determine the weight because it is hard to determine the similarity of different English sentences whether the word item similarity is dominant or the word order similarity is dominant, so is the ratio.

In view of the issue, the lexical chunk theory is introduced into the short English text similarity algorithm to make up for the defects of the keyword-overlap similarity algorithm and its improved algorithm by using the semantic connotation and the fixed word order of lexical chunks.

2.2 Proposed Method

2.2.1 Lexical chunk theory

In the mid-1970s, Becker [17] took the lead in proposing the concept of the lexical chunk. The lexical chunk is a fixed or semi-fixed combination of words that exists between grammar and vocabulary, which can be estimated in advance, thus it is beneficial to the learner's application. Becker argued that the smallest unit of human speech communication was not a single word, and the fluency was not dependent on the number of individual words, but rather the number of words stored in the learner's memory. Nattinger and DeCarrico [18] illustrated that the lexical chunk was actually a combination of words with grammatical form and meaning.

After years of research and development, this concept has been developed into a complete set, which is defined as the lexical chunk theory. The English teaching and English sentence analysis which are based on the lexical chunk theory have been fruitful to some extent. By in-depth analysis of these achievements, the characteristics of the lexical chunk theory in the paper are summarized when applied to the English sentence analysis.

- For the English sentence, the smallest unit of semantic representation is not the word, but the lexical chunk. It is consistent with the law of English language to study the similarity algorithm.
- The lexical chunk can reflect English grammar to some extent. Grammar includes the word order, and the similarity algorithm can ignore the word order research and make the algorithm efficient and easy to be realized.
- The lexical chunk can be estimated in advance. The English sentence can be divided into the lexical chunk in advance. And its characteristics are also the theoretical basis of this algorithm.

2.2.2 Data preprocessing

The similarity algorithm based on the lexical chunk theory requires to calculate the number of each passage and the number of public lexical chunks. Therefore, the data preprocessing of the paper is completed by the lexical chunk division.

Example 2.2. T_1: *A thin brown dog runs by the fat cat.*
T_2: *A thin brown cat runs by the fat dog.*

The data is divided by the key word, which is shown as follows.
T_1: *{ thin, brown, dog, runs, by, fat, cat }.*
T_2: *{ thin, brown, cat, runs, by, fat, dog }.*

The data is divided by the lexical chunk, which is shown as follows.

T_1: *{ thin brown dog, runs by, fat cat }.*

T_2: *{ thin brown cat, runs by, fat dog }.*

2.2.3 LC-SETSA algorithm description

The main idea of the LC-SETSA algorithm is presented in Fig. 1.

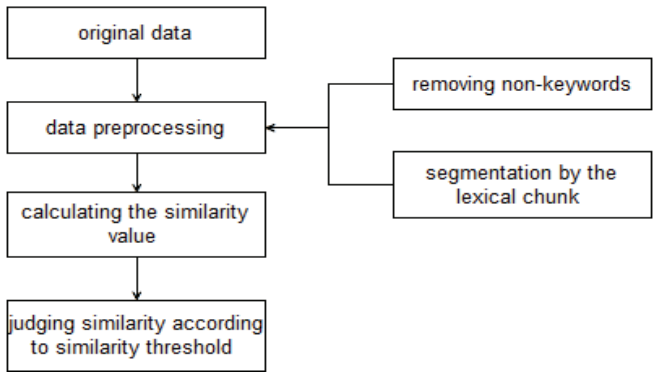

Fig. 1. LC-SETSA algorithm framework diagram.

The LC-SETSA algorithm is proposed on the basis of the keyword-overlap algorithm, and the calculation formula is as follows:

$$Sim_c(T_1, T_2) = \frac{samechunks(T_1, T_2)}{(Alonechunks(T_1) + Alonechunks(T_2))/2} \tag{3}$$

where, $samechunks(T_1, T_2)$ is the number of the public lexical chunks existing in both T_1 and T_2, $Alonechunks(T_1)$ is the number of the lexical chunks existing in T_1, and $Alonechunks(T_2)$ is the number of the lexical chunks existing in T_2.

According to Example 2.2, the text similarity between T_1 and T_2 is calculated as follows:

$$Sim_a(T_1, T_2) = \frac{2\times7}{7+7} = 1, \qquad Sim_c(T_1, T_2) = \frac{1}{(3+3)/2} = 0.33.$$

The calculation results show that the similarity value of the algorithm is 1, and the two sentences are the same. The similarity value is calculated as 0.33 by the similarity algorithm based on the lexical chunk theory, and the similarity of the two sentences is 33%. It is observed that in Example 2.2 the semantics of T_1 are different from those of T_2, and even they are the opposite, but the statements are similar.

3. Results

3.1 Experiment Data

The 400 essays of final public foreign language examination from the university undergraduates are

selected randomly, the sentences about the topic description are selected from the 400 essays, and 200 sentence pairs are randomly constituted as the test data.

A certain number of the college English teachers are selected to artificially determine whether the semantic meanings are similar. Make sure that each sentence is judged artificially by three teachers. If the result is similar, the sentence is marked as 1, otherwise it is marked as 0. Then, a teacher who has been engaged in the research and teaching of lexical chunks is chosen to divide the 200 sentence pairs into artificial lexical chunks. The 200 sentence pairs marked with the similarity identifier and divided into chunks are used as the test samples for these experiments. The method is shown in Formula (4).

$$Mark(T_1, T_2) = \begin{cases} 1, & \frac{M_a+M_b+M_c}{3} > 0.5 \\ 0, & \frac{M_a+M_b+M_c}{3} < 0.5 \end{cases}, \quad (4)$$

where, M_a, M_b, and M_c are the mark to the same sentence pair from the three teachers.

A teacher who has been engaged in the lexical chunk research and teaching is selected to divide the 200 sentences artificially.

The similarity identification is marked, and the 200 sentence pairs with the lexical chunk segmentation are used as the test sample for this experiment.

3.2 Experiment Settings

In all the experiments of this research, the similarity threshold is set to 0.35, which is obtained repeatedly.

Experiment 1

Objective: To verify the feasibility of the LC-SETSA algorithm of the paper in similarity calculation.

Process: The keyword-overlap similarity algorithm and the proposed algorithm are used respectively to make a similarity judgment on the 200 sentence pairs, comparing the number of similar sentence pairs, the number of non-similar sentence pairs and the number of sentence pairs that are judged correctly (assuming that all the artificial judgment results are correct), and the results are analyzed to verify the feasibility.

Experiment 2

Objective: To verify the stability of the LC-SETSA algorithm of the paper in similarity calculation.

Process: The keyword-overlap similarity algorithm and the proposed algorithm are adopted, respectively, to calculate the similarity value of the sentences, and the comparison is carried out between the similarity average values of similar sentence pairs and that of the non-similar sentence pairs, and the results are analyzed to verify the stability.

Experiment 3

Objective: To verify the effectiveness of the LC-SETSA algorithm of the paper in similarity calculation.

Process: The cosine similarity algorithm based on keywords and the cosine similarity algorithm based on lexical chunks (sublimation of this algorithm) are used respectively to make a similarity judgment on the 200 sentence pairs, comparing the number of similar sentence pairs, the number of non-similar sentence pairs and the number of sentence pairs that are judged correctly (assuming that

all the artificial judgment results are correct), and the accuracy of the results is evaluated to verify the effectiveness.

3.3 Experiment Results

The results of Experiment 1 are shown in Table 1.

Table 1. Results of feasibility experiment

Algorithm	Number of similar sentence pairs	Number of non-similar sentence pairs	Number of sentence pairs judged correctly
Artificial judgment	63	137	200
Keyword-overlap similarity algorithm	97	103	147
LC-SETSA algorithm	55	145	189

The results of Experiment 2 are shown in Table 2.

Table 2. Results of stability experiment

Algorithm	Similarity value of similar sentence pairs	Similarity value of non-similar sentence pairs
Artificial judgment	1	0
Keyword-overlap similarity algorithm	0.784	0.362
LC-SETSA algorithm	0.715	0.196

The results of Experiment 3 are shown in Table 3.

Table 3. Results of effectiveness experiment

Algorithm	Number of similar sentence pairs	Number of non-similar sentence pairs	Number of sentence pairs judged correctly
Artificial judgment	63	137	200
Cosine similarity algorithm			
Based on keywords	75	125	173
Based on lexical chunks	58	142	182

4. Discussion

About Experiment 1:

According to the data in Table 1, the judgment accuracy of each algorithm is calculated respectively. It can be perceived that the judgment accuracy of the LC-SETSA algorithm is 83% for the similarity value of sentence pairs, and the judgment accuracy of the keyword-overlap similarity algorithm is 73.5% for the similarity value of sentence pairs. It can also be observed from the data in Table 1 that the judgment results of the LC-SETSA algorithm are closer to the artificial judgment results (real results). The experimental results show that the accuracy of the LC-SETSA algorithm is higher than that of the keyword-overlap similarity algorithm. The experiment verifies that the proposed algorithm is feasible.

About Experiment 2:

It can be seen from Table 2 that the average results of the keyword-overlap similarity algorithm and those of the LC-SETSA algorithm of the paper are 0.784 and 0.715, respectively, which are both higher than 0.35 and closer to 1, with little difference. The average results of the keyword-overlap similarity algorithm and those of the LC-SETSA algorithm are 0.312 and 0.196, which are both lower than 0.35 and 0.196, and closer to 0. Hence, the experimental results verify that the LC-SETSA algorithm has the optimal stability.

About Experiment 3:

According to the data in Table 3, the judgment accuracy of each algorithm is calculated respectively. The judgment accuracy of the cosine similarity algorithm based on keywords is 83% for the similarity values of sentence pairs, and the judgment accuracy of the cosine similarity algorithm based on lexical chunks for the similarity value of sentence pairs is 91%. The experimental results demonstrate that for the similarity algorithm based on word segmentation, the word segmentation method is more favorable than the keyword method. The experiment shows that the proposed LC-SETSA algorithm is effective.

5. Conclusion

Based on the keyword-overlap similarity algorithm, the lexical chunk theory is introduced and accordingly a novel short-text similarity algorithm is designed in this paper, which is referred to as the short English text similarity algorithm based on lexical chunk theory. The LC-SETSA algorithm takes advantage of the characteristics that English lexical chunk lies between word and grammar, and takes the word item information and the word order into account. Examples and experimental results show that it is feasible, stable, and effective to study the text similarity by introducing lexical chunk theory. The defect of the LC-SETSA algorithm is that it ignores the effect of the synonymous lexical chunk on the similarity calculation, which thus is applicable to the similarity calculation of the data with more public lexical chunks. The novel algorithm of the paper is only a preliminary research of the text similarity calculation using the lexical chunk theory. In the future research, the further investigations and explorations are needed for the algorithm improvement.

Acknowledgement

This research was funded by the Education Department of Heilongjiang Province of China (Grant No. 135309463 and 135509118).

References

[1] C. Banea, S. Hassan, M. Mohler, and R. Mihalcea, “UNT: a supervised synergistic approach to semantic text similarity,” in *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval)*, Montreal, Canada, 2012, pp. 635-642.

[2] H. Liang, K. Lin, and S. Zhu, “Short text similarity hybrid algorithm for a Chinese medical intelligent question answering system,” in *Technology-Inspired Smart Learning for Future Education*. Singapore: Springer, 2020, pp. 129-142. https://doi.org/10.1007/978-981-15-5390-5_11

[3] S. Banerjee, S. Kaur, and P. Kumar, "Quote examiner: verifying quoted images using web-based text similarity," *Multimedia Tools and Applications*, vol. 80, pp. 12135-12154, 2021. https://doi.org/10.1007/s11042-020-10270-4

[4] Y. Liu and M. Chen, "Applying text similarity algorithm to analyze the triangular citation behavior of scientists," *Applied Soft Computing*, vol. 107, article no. 107362, 2021. https://doi.org/10.1016/j.asoc.2021.107362

[5] X. Lin, M. Zhang, X. Bao, J. Li, and X. Wu, "Short-text Classification Method Based on Concept Network," *Computer Engineering*, vol. 36, no. 21, pp. 4-6, 2010. https://doi.org/10.3969/j.issn.1000-3428.2010.21.002

[6] C. Jin and H. Zhou, "Chinese short text clustering based on dynamic vector," *Computer Engineering and Applications*, vol. 47, no. 33, pp. 156-158, 2011.

[7] X. Q. Zhao, Y. Zheng, and H. Q. Chu. "Application of concept tree in semantic similarity of short texts," *Computer Technology and Development*, vol. 22, no. 6, pp. 159-162, 2012.

[8] J. Yin, D. Chao, Z. Liu, W. Zhang, X. Yu, and J. Wang, "Model-based clustering of short text streams," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London, UK, 2018, pp. 2634-2642. https://doi.org/10.1145/3219819.3220094

[9] T. Schick, H. Schmid, and H. Schutze, "Automatically identifying words that can serve as labels for few-shot text classification," in *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain, 2020, pp. 5569-5578. https://doi.org/10.18653/v1/2020.coling-main.488

[10] J. W. Sun, X. Q. Lu, and L. H. Zhang, "Short text classification based on semantics and maximum matching degree," *Computer Engineering and Designing*, vol. 34, no. 10, pp. 3613-3618, 2013.

[11] H. T. Nguyen, P. H. Duong, and E. Cambria, "Learning short-text semantic similarity with word embeddings and external knowledge sources," *Knowledge-Based Systems*, vol. 182, article no. 104842, 2019. https://doi.org/10.1016/j.knosys.2019.07.013

[12] G. Majumder, P. Pakray, R. Das, and D. Pinto, "Interpretable semantic textual similarity of sentences using alignment of chunks with classification and regression," *Applied Intelligence*, vol. 51, pp. 7322-7349, 2021. https://doi.org/10.1007/s10489-020-02144-x

[13] Z. Liu, C. Lu, H. Huang, S. Lyu, and Z. Tao, "Text classification based on multi-granularity attention hybrid neural network," 2020 [Online]. Available: https://arxiv.org/abs/2008.05282.

[14] P. Huang, G. Yu, H. Lu, D. Liu, L. Xing, Y. Yin, N. Kovalchuk, L. Xing, and D. Li, "Attention-aware fully convolutional neural network with convolutional long short-term memory network for ultrasound-based motion tracking," *Medical Physics*, vol. 46, no. 5, pp. 2275-2285, 2019. https://doi.org/10.1002/mp.13510

[15] N. Peinelt, D. Nguyen, and M. Liakata, "tBERT: topic models and BERT joining forces for semantic similarity detection," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Virtual Event, 2020, pp. 7047-7055. http://dx.doi.org/10.18653/v1/2020.acl-main.630

[16] R. Zhang, G. Yang, and H. Wu, "A new measure of semantic similarity between unknown Chinese words based on HowNet," *Journal of Chinese Information Processing*, vol. 26, no. 1, pp. 16-21, 2012.

[17] J. D. Becker, "The phrasal lexicon," in *Proceedings of the 1975 Workshop on Theoretical Issues in Natural Language Processing*, Cambridge, MA, 1975, pp. 60-63. https://doi.org/10.3115/980190.980212

[18] J. R. Nattinger and J. S. DeCarrico, *Lexical Phrases and Language Teaching*. Oxford, UK: Oxford University Press, 1992.

Na Li **https://orcid.org/0000-0002-7692-0984**

She received her M.S. degree in Qiqihar University in 2011. Now, she is a lecturer at Qiqihar University. Her research interests include optimization algorithm, artificial intelligence, machine learning, pattern recognition, network security, etc.

Cheng Li **https://orcid.org/0000-0002-9656-5385**

He received his M.S. degree in science and technology of computer from Qiqihar University in 2013. Now, he is currently pursuing the Ph.D. degree in science and technology of computer in Harbin Engineering University. He is a professor at College of Computer and Control Engineering, Qiqihar University. His research interests include artificial intelligence, optimization algorithm, network security, etc.

Honglie Zhang **https://orcid.org/0000-0002-3186-2629**

She received her Ph.D. degree in Harbin Engineering University in 2011. Now, she is a professor at College of Computer and Control Engineering, Qiqihar University. Her research interests include embedded system, artificial intelligence, optimization algorithm, etc.