Journal of Informetrics 18 (2024) 101467

Contents lists available at ScienceDirect

Journal of Informetrics

journal homepage: www.elsevier.com/locate/joi

Enhancing semantic text similarity with functional semantic knowledge (FOP) in patents

Hao Teng [a], Nan Wang [a,b,*], Hongyu Zhao [c], Yingtong Hu [d], Haitao Jin [b]

[a] Beijing Key Laboratory of Internet Culture and Digital Dissemination Research, Beijing Information Science and Technology University, Beijing 100101, China
[b] Computer School, Beijing Information Science and Technology University, Beijing 100101, China
[c] Tencent Technology (Beijing) Co. Ltd, Beijing 100037, China
[d] Huawei Nanjing Research Institute, Nanjing 210012, China

ARTICLE INFO

Keywords:
Semantic text similarity (STS)
Subject-action-object (SAO)
Functional semantic knowledge (FOP)
Bi-LSTM
Patent similarity
Pre-trained embedding

ABSTRACT

The semantic text similarity (STS) estimation between patents is a critical issue for the patent portfolio analysis. Current methods such as keywords, co-word analysis and even the Subject-Action-Object (SAO) algorithms, are not quite reasonable for the patent similarity calculation due to the lack of fine-grained semantic knowledge, "property-parameter" features and flexible "functional or non-functional" combinations. In the meanwhile, standardized similarity datasets are also unavailable. In this paper, we have proposed a new kind of functional semantic knowledge (Function-Object-Property, i.e., FOP) instead of SAO triples, which can contribute directly to enhance the patent similarity. Moreover, patent STS datasets, including the matching dataset and the ranking dataset, have firstly been processed and released as benchmarks for the comparative evaluation. Preliminary results have demonstrated that FOP-based methods are more appropriate in the STS tasks incorporated with IPC codes, weights' assignments and patent pre-trained vectors. To be further, the deep interaction-based models with the averaged FOP embeddings are recommended to be one of the most optimal choices of effectively improving the semantic learning capability. Finally, a new patent similarity calculation framework is summarized and successfully applied in the patent retrieval, which highlight that the proposed methodology serves as a dominant power in diverse patented STS tasks.

1. Introduction

As a major form of intellectual remarks, patents have recorded more than 90 % latest technical portfolio all over the world, of which 80 % would firstly be published (An et al., 2021). In order to fulfill the further analysis, the selection of appropriate matching methods is increasingly vital for the similarity measurement (Abbas et al., 2014; He et al., 2020; Zhang & Pun, 2022). The patent semantic text similarity fully deserves deeper attention as a solid basis of most of patent analysis, such as prior art search, passage retrieval, litigation analysis, technology forecasting and quality evaluation. Moreover, patent semantic similarity is also one of the fundamental necessities since it is achievable to derive technical intelligence efficiently, detect the risk of infringement, evaluate the novelty and innovation of patents (Krestel et al., 2021; Wang et al., 2019). Early-stage patent similarity methods in the literature could be classified into a few

* Corresponding author.
E-mail address: wangnan8848@126.com (N. Wang).

https://doi.org/10.1016/j.joi.2023.101467
Received 23 February 2023; Received in revised form 28 October 2023; Accepted 7 November 2023
Available online 14 November 2023

categories using IPC codes, the bibliographic analysis and lexical-typed approaches (Park et al., 2013). With regards to the content similarity, most of the related methods are the string-based, vector-based and knowledge-based ones (Kwon et al., 2021; Majumder et al., 2016). In recent years, the SAO structures proposed to represent the technical solution itself help to enrich the semantic understanding of patents in the patent similarity (Choi et al., 2011; Wang et al., 2017; Yoon et al., 2013).

However, limitations and problems still exist in SAO-based methods: firstly, most of the SAO extraction algorithms heavily rely on excellent lexical and syntactic models using which the SAO element extraction accuracy in different domains or fields are not quite acceptable. Additionally, the SAO processing is not easy with a great deal of the semantic uncertainty (Sun et al., 2022). The SAO triples themselves make insufficient effects on the subsequent similarity calculation due to the SAO complexity without a universally reasonable formula. The absence of an accepted standard for the similarity evaluation across different fields seems to be neglected. Furthermore, the amount discrepancies of SAO triples in different patents also have intangible implications on the similarity calculation (An et al., 2021; Chen et al., 2020; Wang et al., 2019). The non-functional relations neglected by SAO triples should also be reflected by a preposition or preposition phrase with regards to patented inventions. For example, the "of" is a preposition in the phrase "a feature of any object", which reflects a subordinate relation that cannot be expressed by SAO structures. To remedy these weaknesses, knowledge-based extension tools (such as the WordNet dictionary) are introduced as required supplements for the text matching (Hussain et al., 2020; Li et al., 2020). In addition, some weighting strategies should also be designed and further investigated for the functional necessities' evaluations (Amir et al., 2017; Wang et al., 2019). The fine-grained patent knowledge extraction including the entities and relations are also explored (Chen et al., 2020). A detailed analysis using the "Entity-Relation-Entity" sequences are also recommended with the embrace of semantic relations, and the improved similarity method has also been realized (An et al., 2021). However, the fine-grained semantic knowledge extraction, knowledge combination and similarity computing strategies wait to be fully investigated to construct a more powerful similarity calculation framework for the patents' analysis.

To address the issue of methodological assessments, the patent semantic text similarity (STS) datasets is also necessary. Prior and current research usually relies on the patent classification systems to measure the patent similarity (Arts et al., 2018). To measure the similarity between any two patents, a Jaccard index was calculated by dividing the number of unique keywords as an average value of 0.24 between patent pairs (among the 4386,405 closest text-matched patent pairs). The results are corresponding to approximately 14 common keywords for two patents with an average number of 37 keywords.[1] In order to generate the patent similarity datasets, the data provided by the USPTO's Office of the Chief Economist (OCE) are paid more attention (Whalen et al., 2020). Although datasets are publicly available, it can be seen that there is no more suitable dataset for testing patent similarity, especially for exploring the semantic similarity computation (Suzgun et al., 2022). In fact, domain-specific texts are rarely understood and processed, but patents are naturally characterized by the assessment of novelty which is achieved through a series of substantive examinations. Generally, this part of data has a kind of manually evaluated semantic matching features. Therefore, it becomes the impulsion for us to process this kind of datasets. In addition, ranking datasets should also be provided that to achieve another reasonably supplementary evaluation of the patent similarity methods.

In summary, some aspects account for the patent semantic similarity deemed as a controversial issue: 1) lack of benchmark datasets; 2) insufficient semantic features; 3) unreliable similarity calculation and evaluation. To complement the aforementioned weakness of the SAO-based and other textual methods, the current literature review of patent semantic similarity computation is presented. How to deepen the functional-centric semantic features from patent texts deserves more attention (Guarino et al., 2022). Herein, we firstly make more efforts on the fine-grained knowledge extraction, representation and use from patents, which contribute a lot to the proposed functional semantic knowledge (FOP). Patent STS datasets and benchmarks are also provided in combinations with the extraction of SAO and FOP features. Moreover, deep learning models and pre-trained models should be considered to build a semantic computing framework (Krestel et al., 2021; Yu et al. 2021). The implementation of the FOP knowledge representation using pre-trained embeddings is also introduced, and meanwhile the performances of deep learning models are also discussed in details. Finally, the optimally selected patent semantic similarity framework is summarized and case studies help verify the proposed method in the simulated patent retrieval.

2. Related works

2.1. Patent semantic features

The common patent features contain Patent ID, IPC codes, Application Date, Assignee Name, and Assignee Country, and patent content indicators are also combined with the help of citation analysis and qualitative semantic analysis of texts (Shih et al., 2010; Wang & Cheung, 2011). Similarity calculations performed with a manual classification scheme or judgment of empirical analyses with a diverse set of expertise is often indispensable in the interdisciplinary research (Leydesdorff et al., 2012; Wang & Liu, 2022). Citation analysis is derived from citation and co-citation relationship inherent in the data (Liu, 2013; Rodriguez et al., 2015). Later, the semantic analysis has gradually been developed with the keywords-based approaches to identify technological components that require expertise knowledge (Choi & Hwang, 2014; Feng et al., 2020). Besides, co-words analysis provides a tool to quickly understand textual meanings. However, disadvantages also exist: (1) homonyms and synonyms of words and terms result in ambiguous interpretations; (2) high-frequency or common terms fail to be discriminated between relevant and irrelevant topics (Yang et al., 2017b). Some tools

[1] https://dataverse.harvard.edu/dataverse/patenttext.

capable of analyzing unstructured information, such as SAO-based mining techniques are proposed instead of keyword-based and indicators' approaches (Park et al., 2012; Wang et al., 2017; Yang et al., 2017a). To be further, a property-function based patent network (PFPN) was outlined to gain the insights about the technological trends (Yoon & Kim, 2012a). Another SAO-TRM uses the Product-Function-Technology (PFT) maps to assist in the decision making (Park et al., 2013). Preliminary literature also utilized the TRIZ-based knowledge to guide semantic analysis of patents (Cong & Tong, 2008; Spreafico & Russo, 2016). Furthermore, there are also product design-oriented functional model definitions and open source libraries of related product-component related functional flows (Cascini & Zini, 2008; Fiorineschi et al., 2020). Recently, deep-learning models are used for entity identification and semantic relation extraction (Chen et al., 2020). Now there are few systematic discussions about the functional semantic features of patents related to the similarity calculation.

2.2. Textual similarity algorithms

Similarity tasks can be divided into four types: paraphrase identification, semantic textual similarity (STS), natural language inference, and question answering. Paraphrase Identification (PI), which identifies whether two sentences express the same meaning (Meek, 2018; Rajpurkar et al., 2016). In addition to feature-level designs, textual similarity has to be algorithmically considered, known as Semantic Textual Similarity (STS) (Dagan et al., 2006). Research predominantly focused either on the document similarity or the word similarity (Saríc et al., 2012) . STS measurement methods play an important role in many applications within natural language processing (Prakoso et al., 2021), which measures the degree of equivalence in the underlying semantics of paired snippets of text (Agirre et al., 2016). The present works upon the determination of text-similarity has been partitioned to three type of approaches such as lexical-based, vector-based and knowledge-based similarities. Lexical-based algorithms namely the character-based and term-based similarity, contain a series of classical methods: N-gram, Longest Common Substring (LCS), Dice's coefficient and Jaccard similarity (Kohila, 2016). Recent developments in machine learning techniques have led to the development of the vector-based methods in a vector space model (Hain et al., 2022; Younge & Kuhn, 2016). Some methods to determine similarities based on statistical probability calculations (Quan et al., 2010), and more nuanced topic models have been developed, such as the latent semantic indexing (LSI) and the Latent Dirichlet allocation (LDA) that assigned weighted probabilities on topic terms to measure document/sentence similarity (Jang et al., 2016; Xu et al., 2021). Deep models represented by supervised learning are now becoming the mainstream (Lan & Xu, 2018). Knowledge-based methods are very helpful as they comprise a highly structured information of words and their meanings (Das et al., 2014) . In order to calculate the semantic similarity, the WordNet is usually chosen as the lexical database of English words for computing the semantic similarity of concepts (An et al., 2021; Miller, 1995). The syntactic patterns and parser trees of contents are also associated with knowledge characteristics (Inan, 2020). The semantic complexity of a sentence can also be encoded with the POS tagging words (Xu et al., 2021). Nowadays, hybrid approaches which combine knowledge-based and vector-based approaches, are gradually popular and helpful for leveraging both semantic methods.

2.3. Deep semantic matching

With the introduction of pre-trained embeddings, deep learning methods have become more acceptable for enabling the dense-vectored analysis of patents with proposed deep neural network architectures (Devlin et al., 2019; Krestel et al., 2021; Viji & Revathy, 2022). Deep matching models have been proposed and divided into Siamese-based methods and interaction-based categories (He & Lin, 2016; Lyu et al., 2021). The early representative Siamese-based model is the DSSM model which is composed of a simple and intuitive bilateral encoded architecture (Huang et al., 2013). Then the proposed CDSSM model further introduced the "word--hash+CNN" structures, which had advantages over others with the simplicity and generalization of the networks (Shen et al., 2014). Recently, some have explored weighted strategies between dependent syntactic structures in contents (Quan et al., 2019). Furthermore, a hybridized approach using the weighted fine-tuned BERT model with the Siamese Bi-LSTM module is implemented (Viji & Revathy, 2022). In contrast, the interaction-based approaches enable the phrase alignments and interactions between the paired contents. Combined with local matching relationships and intrinsic hierarchy of contents, the dot product based networks were highlighted using topic models. The ARC-I/ARC-II models were proposed by means of CNNs and gate mechanisms to complement the long sequence representations (Hu et al., 2014). Furthermore, a new deep architecture named MV-LSTM has been presented to match two sentences with multiple positional sentence representations using the Bi-LSTM layers (Wang et al., 2015). Besides, the Bilateral Multi-Perspective Matching (BIMPM) and the Densely-connected co-attentive Recurrent Neural Network (DRCN) models had also achieved state-of-the-art performances on most of STS benchmark datasets (Lan & Xu, 2018). Besides, a novel five-layer neural network called CapsTM, has employed the capsule networks (Yu et al., 2021). Interactive attention networks are adopted to dynamically generate the matching matrix to achieve new content representations (Zhao et al., 2020). Now neural networks that cater to the STS tasks have more preferences of the end-to-end model architectures (Giabelli et al., 2022). The models are mainly designed using the following modules: Input Embedding Layer; Context Encoding Layer; Interaction and Attention Layer; Output Classification Layer (Raj et al., 2022). In a word, lexical taxonomies, multi-knowledge interactions and multi-dimensional computations contribute a lot to the semantic computing.

3. Methodology

In order to reasonably use and evaluate the proposed methodology, we have designed a complete flowchart shown in Fig. 1. This covers a few aspects: 1) the STS datasets preparation; 2) The functional semantic ontology of patents is mainly proposed in favor of the

FOP knowledge extraction; 3) Sequence labeling models (i.e., named entity recognition (NER) models) are trained based on the annotated patent corpus to obtain the functional semantic knowledge (FOP structures); 4) Different pre-trained language models with FOP embeddings for subsequent experiments are prepared and discussed; 5) Experiments with text similarity methods and especially deep matching models are developed, and the evaluation metrics of the matching patent dataset and the ranking patent dataset are given, respectively.

3.1. STS dataset processing

A major challenge in the patent similarity research is the lack of the semantic text similarity (STS) corpora. Nowadays, bulk data from Patent Trial and Appeal Board (PTAB) assembled with patent appealing datasets, such as the due diligence, litigation and infringement files, are also distributed online.[2] As is known to all, referenced patents from most of these PTAB appealing files are used to assess the novelty and creativity of patents at the semantic similarity level, which are in nature appropriate for exploring the semantic discrepancies between paired patents. Herein, we have tried our best to manipulate the real patent STS datasets from the PTAB files and patent dumps step by step.

Firstly, a series of crawlers for scanning and downloading decision files from the PTAB website[3] were designed and developed, and then imprecise downloaded metadata should be necessarily cleaned and curated to text formats. Secondly, the public numbers of targeted and referenced patents mentioned in the decision files were obtained, and the corresponding patent metadata were also searched and then fully processed from the US patent and trademark office (USTPO), European patent office (EPO) and World Intellectual Property Organization (WIPO). The patent metadata were pre-processed by lowercasing the texts, tokenizing all words, and eliminating stop-words. Next, the related patent documents, especially the independent claims and examination opinions of "targeted and referenced" patents, were also extracted. The subsequent way is to select, assess and determine how similar or dissimilar patent contents are from one another. Generally speaking, the semantic similarities are often determined by patent claims themselves, and claim content sections have been utilized in a variety of related tasks such as the patent retrieval, litigation or recommender systems (Lee et al., 2013). Furthermore, the claims are also separately split to facilitate the weights' assignments for key features and normal descriptions of patents. Lastly, the number of processed patent pairs is up to 8770, and the number of independent patents is 12,758. To be specific, the dataset of paired patents is divided into A, G and H IPC groups, as shown in Table 1.

As a result, two types of patent STS datasets are proposed. One is the patent matching dataset which aims at providing the similarity evaluation capability of the paired inventions. As shown in Table 2, a similarity criteria of the 3-point scale is selected where a score between 0 and 3 will be given for each patent pair on the basis of experts' judgments of the patent semantic similarity. For example, patent pairs involved with infringement appeals and the same former two-IPC codes are given a score of 3, which indicates a relatively high semantic similarity between the two patents. The zero score means that there is none of any similarity (different IPC codes, no infringement, few common keywords) between two patents randomly selected from this patent dataset. Another ranking dataset is also further processed for the easy use of the ranking similarity evaluation in the retrieval scenes. Each set of patents consists of a paired patents from the entire dataset and one hundred of patents randomly selected with the same first two-IPC classification codes of paired patents from the USTPO bulk data. In the evaluation process, the average ranking values of recalled objective patents among each set of patents, is statistically listed and then comparatively used to investigate the effectiveness of different similarity calculation methods. The ranking indexes such as the "hits@N", MAP and NDCG indexes, are usually employed in the academics.

3.2. Functional semantic ontology construction

Semantic features in patent contents involve a variety of fine-grained knowledge as well as relevant auxiliary technical descriptions. Conventional SAO-based methods cannot be accepted with a high SAO element extraction accuracy, which accounts for a great deal of the semantic uncertainty (Sun et al., 2022). The weights' assignments, amount discrepancies and calculation formula of SAO triples in different patents have intangible implications on the similarity calculation.

According to the actual patent analysis and modernized TRIZ guidance, a patent can be decoupled into a complete technical solution with regards to the technical method or product improvements. The technical fields or the products involved in patents are often also mentioned in the technical background, which can be obtained by simple rules or algorithms. To be further, the key features of technical contents should be described in a functional hierarchical structure, including various types of knowledge entities and corresponding relationships. In fact, current patent ontologies are not quite reasonable for the patent semantic analysis because of the lack of "property-parameter" knowledge and flexible "functional or non-functional" combinations. Recently, a promising patent information extraction framework is proposed where deep-learning models are used for entity identification and semantic relation extraction, respectively (Chen et al., 2020). The achievements of the functional analysis have evolved and led to the creation of entity and relationship characteristics in patents. Herein, the functional semantic ontology of these technical solutions is proposed in Fig. 2 (a), which helps any patent to be decomposed into a series of functional combinations in convenience of the whole semantic understanding. Apart from the basic descriptions of "field-tech-product" types of knowledge, the core of the general invention lies in the patent claims (especially the independent claims), and therefore the relevant functional semantic contents would be elaborately described and represented at a deeper semantic level by the "functions, objects and properties" entities and their relations (i.e., the FOP

[2] https://www.uspto.gov/patents/ptab/about-ptab
[3] https://developer.uspto.gov/ptab-web/#/search/decisions

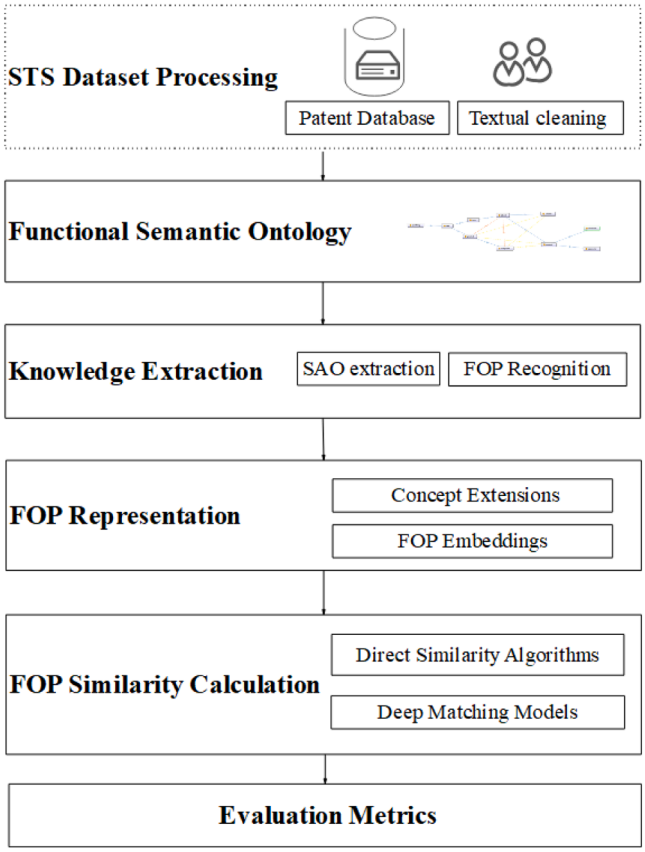


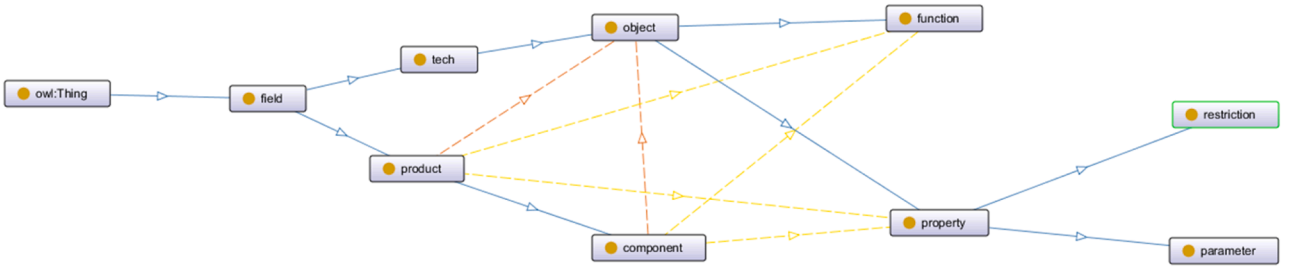
Fig. 1. The flowchart of the whole methodology.

Table 1
Details of paired patents in the STS datasets.

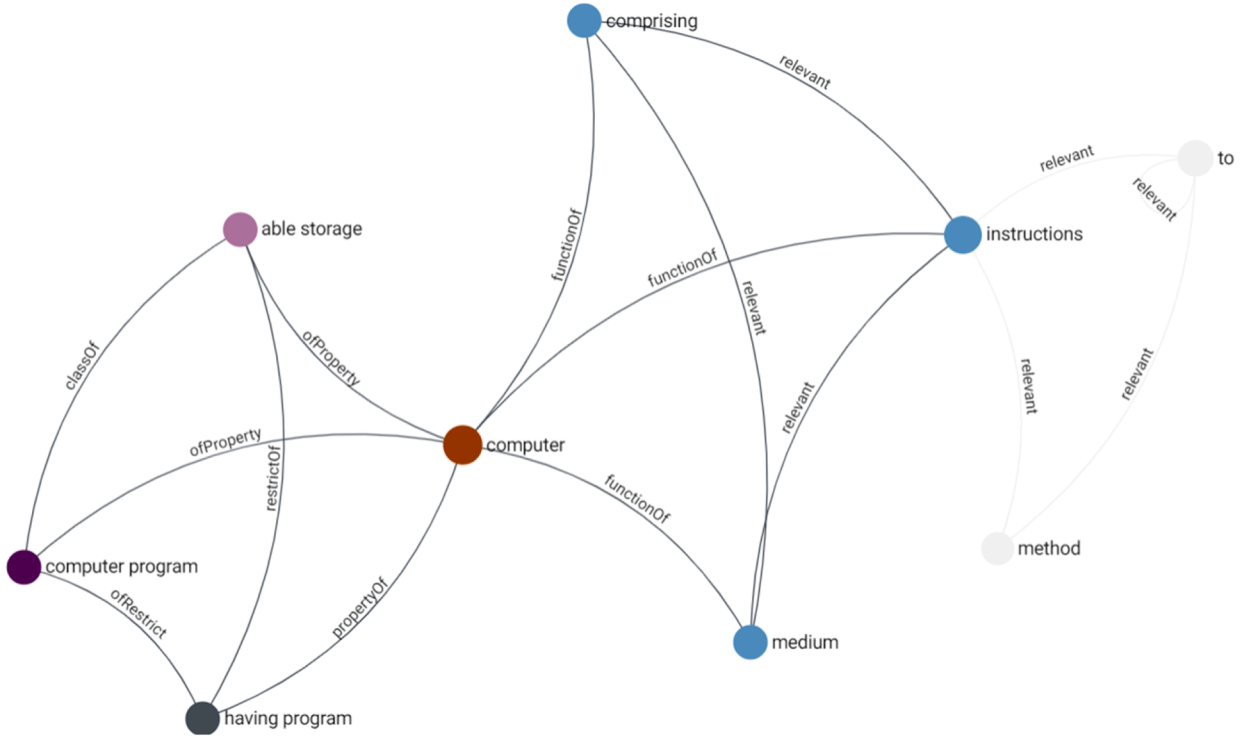
Rank	IPC	Explanations	Count
1	G06F	Electric Digital Data Processing	4713
2	G06Q	Information And Communication Technology [ICT]	1556
3	H04L	Transmission Of Digital Information	829
4	H04N	Pictorial Communication	750
5	A61K	Preparations For Medical, Dental Or Toiletry Purposes	696
6	A61B	Diagnosis; Surgery; Identification	664
7	H01L	Semiconductor Devices	502
8	H04W	Wireless Communication Networks	353
9	H04B	Transmission	295
10	A61F	Filters Implantable Into Blood Vessels; Prostheses	275

Table 2
The similarity assignment reference criteria with a 3-point scale.

Similarity values	Similarity assignment reference
3	Appealing (such as infringements or litigation or due diligence) patent pairs with the same first two-IPC codes.
2	Patent pairs with appeals or infringements or litigation or due diligence but different IPC codes.
1	A sample of non-appealing data with the same first two-IPC codes, and satisfies one of the aspects: the same main technical keywords, the same IPC codes or other classifications; citation relationships.
0	None of any similar features discussed above.



(a) The functional semantic ontology of patents.



(b) A simple graphical view of the functional semantci knowledge (FOP) of a patent.

Fig. 2. Decoupling of an invention using the functional semantic ontology and the FOP knowledge.

structures), as shown in Fig. 2 (b). These FOP-type knowledge involved in the specific patents can be labeled and extracted in a sequence by the named-entity-recognition (NER) methods. Moreover, the relationships among entities should also be classified by the relation extraction (RE) methods.

3.3. FOP extraction and representation

As a previous step before the functional semantic knowledge processing, the crawled and archived datasets should be firstly pre-processed. The meaningless characters or noises are firstly removed, such as many HTML tags, and the steps are listed in Table 3.

In the following, the FOP knowledge extraction as a text annotation task actually requires the NER labeling of different types of the functional semantic knowledge. Considering that conventional annotation methods require more feature engineering, that is to say, effective features are difficult to obtain. In contrast, deep neural network structures can be used to automatically encode relevant features. Herein, the classical Bi-LSTM+CRF model is adopted for the FOP extraction, as shown in Fig. 3 (a). The model structure consists of three main layers: the word embedding layer, the bi-directional long short-term memory network (Bi-LSTM) layer, and the conditional random field (CRF) layer. The word embeddings are from the BERT/ALBERT pre-trained models, and of course, the word2vec or Glove embeddings are also considered as substitutes. Due to the long-sequence dependency of patent contents, the feature representation using the bi-directional phrase embedding can better encode the contextual information and help improve the model effects. Therefore, the Bi-LSTM module is further designed and regarded as an encoding layer to output the label probability distribution corresponding to the current candidates. The output from the Bi-LSTM layer is used as the input of the CRF layer. Conditional Random Fields (CRF) is the commonly used model with a high accuracy in the entity labeling. Further access to the conditional random field layer, given a random variable X as the probability distribution of encoded features. The distribution of another labeling result Y is predicted by the transfer probability in the CRF. The model gets the corresponding "B-I-O" triadic labels for the FOP entity extraction. The FOP phrases are inferred by designating their first and last character boundaries of "B-I-O" labels. Part of training datasets are manually labeled by the BRAT tool.[4] FOP definations are derived from a technical system, which are divided into three types:

Function: behavior, operate or effect.
Object: substance, flow, state, material or other matter entity.
Property: entity feature or parameter.

As for the labeling and annotation rules, our team consisting of six members spends almost 6∼12 months on the data labeling. The corresponding author is responsible of allocating datasets to each annotator and reviewing the annotated results by each annotator. All annotators are firstly trained by acknowledging the FOP definations and annotation rules. We have iteratively edited the patent corpus, and put more focus on the FOP annotation clues. The qualified labeled patents are crossly validated and stored directly in archived files. Then the trained extraction models could be incorporated for the dataset generation. With millions of global patents employed, the extraction models are iteratively trained and FOP thesaurus (i.e., DICT) are also constructed for the subsequent evaluation. The FOP thesaurus (i.e., DICT), parts of speech, sentence parsing (i.e., RULES) help enhance the FOP extraction results. In Fig. 3 (b), the FOP extraction results are comparatively evaluated in the large amount of labeled corpus (about 2.5 million of datasets uniformly distributed in IPC codes (A-H groups, where C group is not covered), and labeled entities can be depicted with the result of {'FUNCTION': 2610417, 'OBJECT': 3827718, 'PROPERTY': 1887742}). The F1 values of FOP entity identification results using BERT+Bi-LSTM + CRF + DICT + RULES are almost above 90 %, and this extraction logic can be verified by the following experiments. Combined with the sufficient FOP knowledge identification in patent sequences, various adjacent combinations of FOP structures (such as FO, FP, FOP, OP, OFP, etc.) can be grouped for the subsequent applications according to the dependency parsing of patent contents.

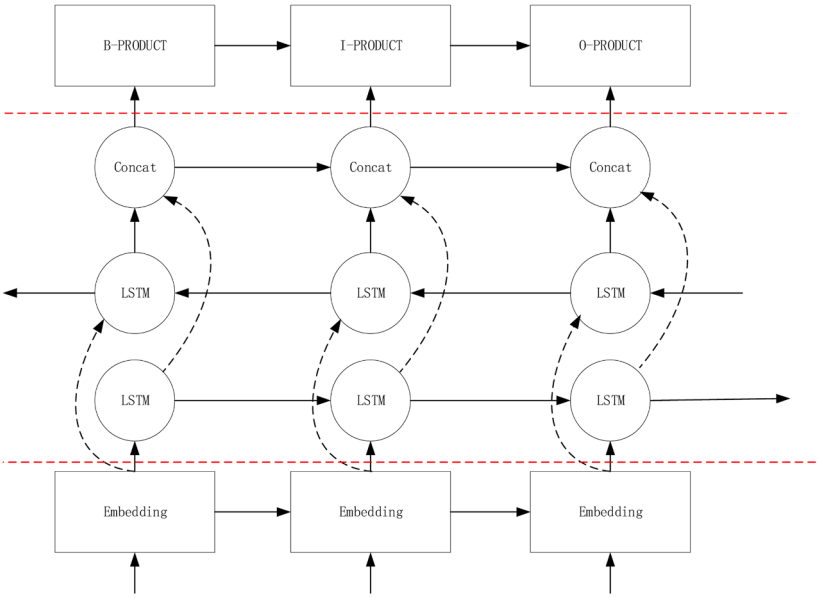
To be further, the SAO triples are also meanwhile processed for the following comparative experiments. With the open-source NLP tools increasingly growing, some general NLP techniques, such as "Stanford CoreNLP", Ollie, Reverb or Knowledgist2.5™ tools, are utilized in the SAO processing (Manning et al., 2014; Yang et al., 2017a; Yoon & Kim, 2012b). To fulfill the collections of SAO triples in a sequence as much as possible, a series of script codes are developed on the Stanford-NLP syntactic dependency parser and Ollie tools to achieve the triadic SAO acquisition and cleaning. Lastly, the SAO triples and FOP knowledge of patents are readily prepared to be fully explored in the following experiments.

The FOP knowledge should be reasonably transformed to accommodate vectored representations corresponding to the meaning of FOP combinations. As mentioned above, these vectored representations or word embeddings can be pre-trained using language models such as the Word2vec, Global Vectors for Word Representation (Glove), Embeddings for Language Models (ELMo) or BERT (Mikolov et al., 2019). The major advantage of pre-trained embeddings is the ability that they could efficiently absorb the word semantics within the FOPs. To be more specific, the knowledge embedding techniques indicate to represent entire semantic information of FOPs. Although there exist a variety of knowledge embedding techniques for directly obtaining vector representations of FOPs (Ji et al., 2022), we have simply adopted the combinations of pre-trained F/O/P element embeddings as FOP representations. Herein, in this paper, the semantic vectors pre-trained from global wiki-dumps and patent corpus are both constructed using Glove models (Pennington et al., 2014). The global wiki-dump corpus consists of varying Wikipedia dumps between the year from 2010 to 2014. In the

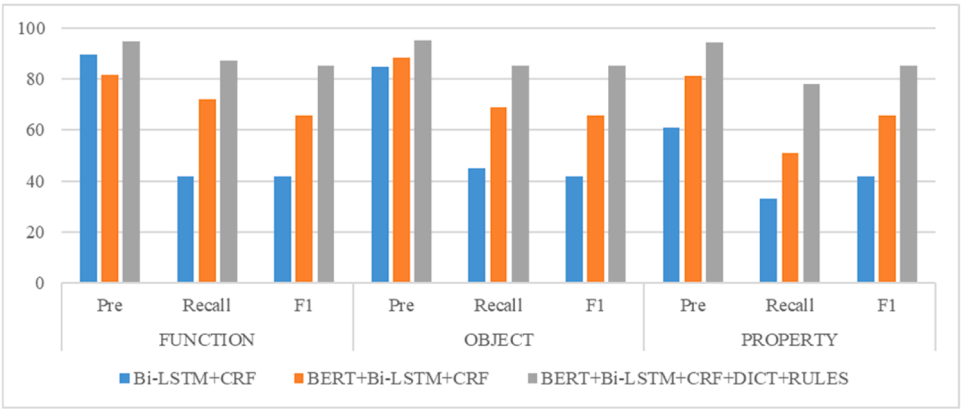
[4] http://brat.nlplab.org/

Table 3
The pre-processing steps of the archived patent datasets.

Processed text	Methods
General HTML tags' processing	Use regular expressions to filter out strings wrapped in </>
HTML tags containing complex browser styles	The decomposed method of the Beautiful Soup library recursively removes all tags that match the criteria, and filters out the style attributes as well
String processing after the md5 hash	Design rules to filter out unordered texts using regular expressions
Illegal Unicode-type characters	Designing rules and regular expressions for filtering Use the Unicode-data normalization



(a) **Schematic diagram of Bi-LSTM+CRF model structure**



(b) **FOP entity identification results using different models and logics**

Fig. 3. FOP knowledge extraction and comparative analysis.

next step, the FOP representations can be obtained by two types: one is the splicing type in which each dimension vector of FOP structures are joined together (if FOP missing then add the zero), and the other is the averaged type where the average value is calculated from F/O/P vectors in each dimension, as shown in Table 4. Then the representations are used for the subsequent similarity calculation.

3.4. FOP similarity calculation

The similarity calculation methods are one of most important sections in the STS tasks. The conventional and deep matching methods are all collected and help effectively improve the patent similarity results. The details are described as follows.

3.4.1. Direct similarity calculation

Direct similarity methods contain the Lexical-based, vector-based and knowledge-based methods. The lexical methods cover the Dice, Jaccard, Inclusion index formulas (Majumder et al., 2016). The vector-based methods are based on the word vector representation in a VSM model of each word, and the distance between the two-word vectors is calculated directly using the Euclidean, Pearson, Spearman and Cosine similarity. Knowledge-based methods are based on the external linguistic knowledge, such as the WordNet dictionary. As for the similarity of two words, the concepts of words are finally used to the calculation, such as the Jiang-Conrath, Lin or Resnik methods. Similarity methods are mainly divided into three types: lexical-based (or string-based), vector-based and knowledge-based algorithms. The methods are listed as follows:

- **Lexical-based Methods**

Common lexical-based methods contain the Dice, Jaccard, LCS (not used here) and Inclusion indexes: each word is counted as a unit after the phrase is separated by spaces: Set (cyber physical system) = (cyber, physical, system). Here are some brief descriptions for calculating the similarity of two phrases. Sometimes the metrics can use WordNet as a knowledge source. The A and B are sets, and |X | and |Y | are the number of elements in the two sets. S1 and S2 be described as text strings.

Jaccard Index. It is also known as the Jaccard similarity coefficient, which is a statistic used to find the similarity and diversity between two finite sets. It is defined as follows:

$$J(\mathrm{A},\mathrm{B}) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| + |A \cup B|} \tag{3.1}$$

The Inclusion Index. It is also called the Szymkiewicz-Simpson coefficient, which is equal to one when set A is a subset of set B.

$$\mathrm{I}(A,B) = \frac{|A \cap B|}{\min(|A|,|B|)} \tag{3.2}$$

The Dice index is defined as:

$$QS = \frac{2|X \cap Y|}{|X| + |Y|} \tag{3.3}$$

where |X| and |Y| is the number of elements in two sets and QS is the quotient of similarity and ranges between 0 and 1. When it is used to measure the similarity of S1 and S2 strings, the coefficient can be calculated as bigrams as follows.

$$sim = \frac{2n_t}{n_{s1} + n_{s2}} \tag{3.4}$$

Wheren_t is the bigram count of the strings andn_{s1},n_{s2} is the number of bigrams in the strings S1 and S2.

- **Vector-based Methods**

Table 4
FOP combinations' representation results using the averaged type.

Combination type	Averaged Vector
OP	$V_i = \frac{0 + V_{Oi} + V_{Pi}}{3}$, $i = 1, 2, 3\ldots 299, 300$
FP	$V_i = \frac{V_{Fi} + 0 + V_{Pi}}{3}$, $i = 1, 2, 3\ldots 299, 300$
FO	$V_i = \frac{V_{Fi} + V_{Oi} + 0}{3}$, $i = 1, 2, 3\ldots 299, 300$
OFP	$V_i = \frac{V_{Oi} + V_{Fi} + V_{Pi}}{3}$, $i = 1, 2, 3\ldots 299, 300$
FOP	$V_i = \frac{V_{Fi} + V_{Oi} + V_{Pi}}{3}$, $i = 1, 2, 3\ldots 299, 300$

The vector representation of each word in a VSM model is calculated separately and finally summed to become the final vector of a phrase. The distance between the two vectors is calculated directly using the Euclidean, Pearson, Spearman and Cosine similarity. Let $A = \{A_1A_2A_3...A_n\}$ and $B = \{B_1B_2B_3...B_n\}$ be n-dimensional vectors, which can also be used to represent a point (a phrase) in space.

Cosine Similarity: It is a similarity measure of two non-zero vectors of an inner product space, which finds cosine of the angle between them. The cosine similarity cos (θ) of two vectors A and B is

$$\cos(\theta) = \frac{A \cdot B}{||A||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{3.5}$$

WhereA_i andB_i is the component of A and B. The cosine similarity of two vectors having the same orientation is 1, and vertical vectors have a similarity of 0.

Euclidean Distance: It is a distance between two points in the Euclidean space. The Euclidean distance between two points A and B is

$$d(A,B) = d(B,A) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2} \tag{3.6}$$

WhereA_i andB_i is the component of A and B.

Pearson's coefficient: We use the Pearson's coefficient as the evaluation metric to measure the correlation between the predicted scores and the gold -set scores of the STS benchmark:

$$r(A,B) = \frac{\sum_{i=1}^{n} (A_i - \overline{A})(B_i - \overline{B})}{\sqrt{\sum_{i=1}^{n} (A_i - \overline{A})^2 \sum_{i=1}^{n} (B_i - \overline{B})^2}} \tag{3.7}$$

where *r* (*A, B*) indicates the correlation between predicted score *A* and gold-set score *B*. Then$\overline{A}$ and$\overline{B}$ are the sample mean of *A* and *B* values. The strength of this correlation can be assessed as follows:

- Very strong: [0.80,1.00]
- Strong: [0.60,0.79]
- Moderate: [0.40,0.59]
- Weak: [0.20,0.39]
- Very weak: [0.00,0.19]

Spearman coefficient: Suppose two random variables are *A*, and *B* (also can be regarded as two sets), and the number of their elements is n. The i th ($1 <= i <= n$) value taken by the two random variables are denoted by A_i and B_i respectively. Sort *A* and *B* (both in the ascending or descending order) to obtain two sets, where elements a_i and b_i are the rows of A_i in *A* and B_i in *B*, respectively. The elements in the sets are correspondingly subtracted to obtain a rank difference set d, where $d_i = a_i\text{-}b_i$, $1 \leq i <= n$. The Spearman coefficient between the random variables *A* and *B* can be calculated from a, b, or d.

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \tag{3.8}$$

Calculated from the ranked sets a and b, the Spearman coefficient is also considered to be the Pearson's coefficient of the two random variables that have been ranked, and the following actually calculates the Pearson's coefficient of a and b).

$$\rho = \frac{\sum_{i=1}^{n} (a_i - \overline{a})(b_i - \overline{b})}{\sqrt{\sum_{i=1}^{n} (a_i - \overline{a})^2 \sum_{i=1}^{n} (b_i - \overline{b})^2}} \tag{3.9}$$

- **Knowledge-based Methods**

Knowledge-based methods are based on the external linguistic knowledge, such as WordNet (Miller, 1995). As for the similarity of two words, the concepts of words are finally used to the calculation. Let c1 and c2 be two concepts.

Leacock-Chodorow. This measure of Leacock-Chodorow similarity is on the basis of the shortest path that connects the two concepts and the maximum depth of the similarity is quantified by

$$Sim_{lch}(c_1, c_2) = -\log \frac{length(c_1, c_2)}{2D} \tag{3.10}$$

where the length of the shortest path between two concepts is node-counted, and D is the maximum depth of the concept taxonomy.

Resnik. The Resnik similarity is based on the information content of the LCS index (Least Common Subsumer, the most specific ancestor node), identified by

$$Sim_{res}(c_1, c_2) = -\log \Pr[LCS] \tag{3.11}$$

where Pr [c] is the probability of an instance of concept c appeared in a large corpus.

In the further step, there remains that how to calculate the similarity between paired FOPs. Different FOP nodes, such as functions, objects and properties, should be interactively matched based on the direct similarity computation or deep matching computation. A unified framework of this solution is figured in Fig. 4. As for any paired elements of FOP structures, the similarity relies on the direct matching methods among elements, their extension phrases and the pre-set threshold value. If one element has been encountered in the paired element's synonyms, the result is regarded as one if the threshold is also complemented. Additionally, the binary and ternary matching situations between FOP elements can also be self-determined according to the requirements. Of course, the threshold values would be variable with regards to the "Function, Object, Property" types. The non-functional FOP combinations are also taken into account using the similarity framework. Considering the background and core technical features, the weights are also assigned on the paired FOPs in different distributions, separately.

3.4.2. Deep matching calculation

As mentioned above, the deep matching models can be divided in two types: Siamese-based and the interaction-based models, and preliminary research demonstrate that deep models can help enhance the semantic representations as well as the automatically comprehensive assignments of feature weights (Lan & Xu, 2018). The following describes a series of the mainstream model architectures among which the DSSM and CDSSM are both the Siamese-type models and others belong to the interaction-type. The DSSM is mainly composed of the embedding layer, multi-layer projection and similarity computation. The DNN is used to map high-dimensional sparse features into dense representations in a semantic space. The final latent semantic vector is obtained for the matching tasks. To remedy the context-missing weaknesses of the DSSM, the CLSM (convolutional latent semantic model) came into beings, which is also known as CNN-DSSM or CDSSM. The difference between CDSSM and DSSM is mainly located in the input layer and the presentation layer. The CDSSM adds trigram features to the input layer using the sliding window from the convolution layer, and furthermore extracts and effectively retains the global context information through the pooling layer.

Some representative interaction-type models are illustrated as follows. The BIMPM is located at the " Matching layer- > Aggregation layer". The use of two-way multi-angle matching and the matching-aggregation structure is the highlight of this model in Fig. 5. In contrast, ARCII model utilizes a CNN layer with multiple convolution kernels of different sizes to obtain multiple N-gram level representations of bilateral inputs, as shown in Fig. 6. Then the interaction matrix based on each N-gram level representation can be calculated. However, the weaknesses of the CNN models have been enlarged with local feature missed. The DRCN model adopts a more diversified interaction strategy in the interaction stage, as shown in Fig. 7. The auto-encoder (AE) is also used for the dimensionality reduction and result regularization. The effectiveness of the interaction is finally maintained in the improved matching accuracy. In the ESIM, the spliced word vectors are transformed by the Bi-LSTM layer, and then local inferences are implemented using intra-sentence attention mechanisms to further represent global inferences of bilateral inputs. This model combines the decomposable attention and multiple forms of interactions encoded by the Max-Pooling and Avg-Pooling module, as shown in Fig. 8. As a result, the semantic representations of bilateral inputs are effectively obtained.

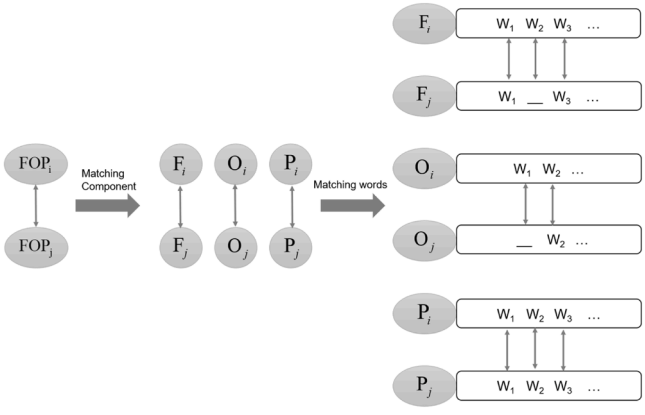


Fig. 4. The process of the FOP similarity calculation.

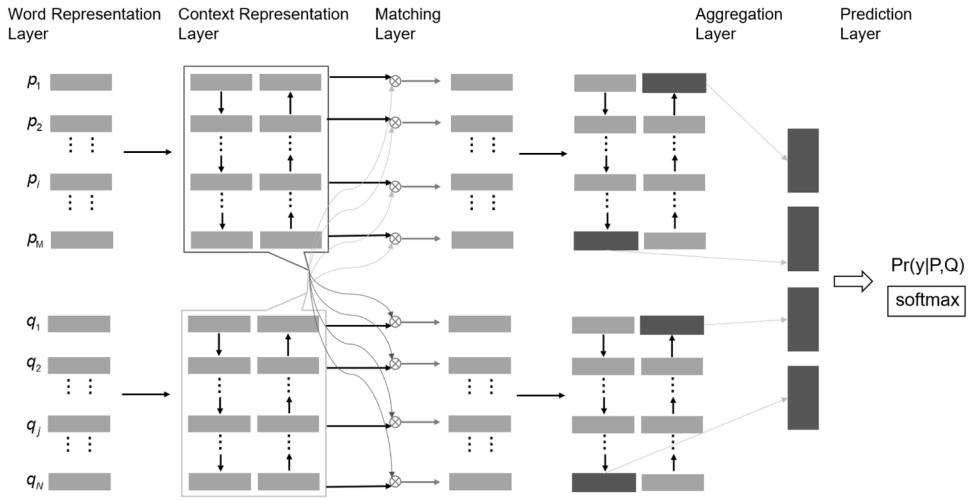


Fig. 5. The BIMPM model structure.

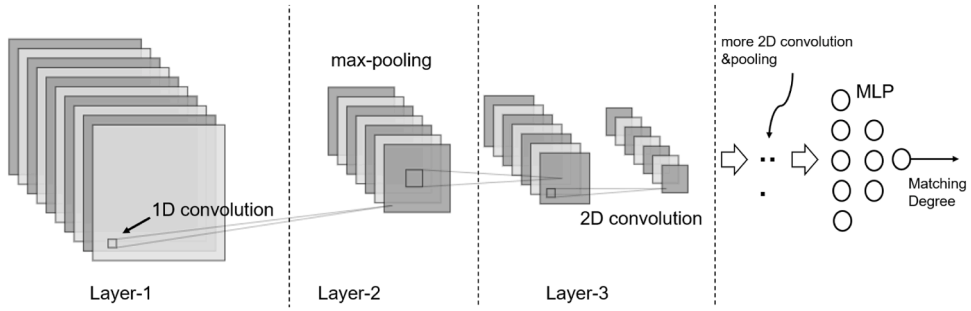


Fig. 6. The ARCII model structure.

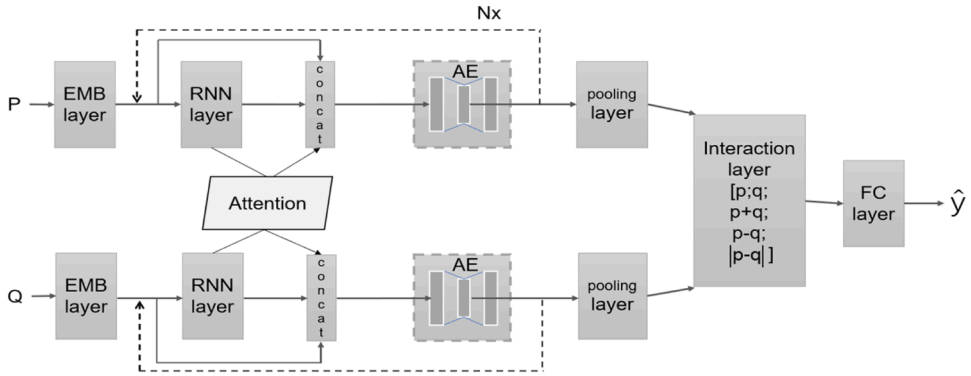


Fig. 7. The DRCN model structure.

3.4.3. Evaluation metrics

The direct similarity measure of the datasets selected in the experiments is one of the best metrics to locate the objective patents by calculating the statistically averaged similarity values of different methods. With regards to the ranking dataset, the average rank of similarity results, such as 10hits@N of patent pairs, is the first necessity. In the ranking tasks, topN@100 is appropriate to be adopted in place of hits@N. Other indexes are also herein used.

Mean Average Precision (MAP):

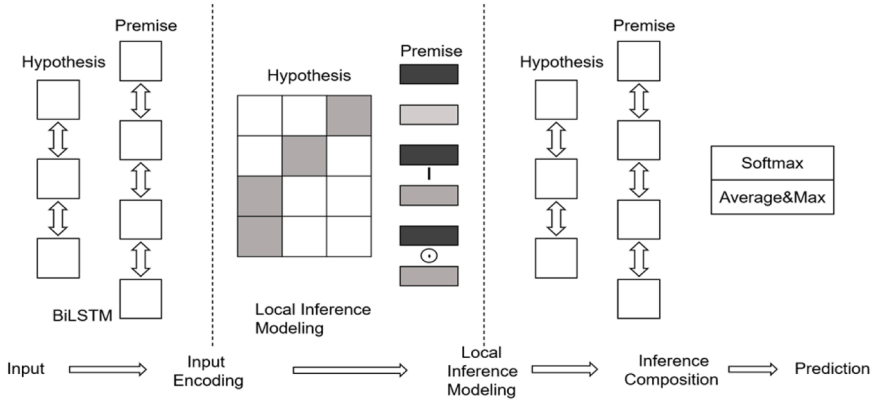


Fig. 8. The ESIM model structure.

$$MAP = \frac{1}{Q}\sum_{q=1}^{Q} AveP(q) \tag{1}$$

Where Q represents the number of queries, and *AveP* (*q*) is the average accuracy of *q* queries.
Mean Reciprocal Rank (MRR):

$$MRR - \frac{1}{Q}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \tag{2}$$

Where *Q* represents the number of queries, and $rank_i$ is the position of the first correct result in the returning list for the *i* query.
Normalized Discounted Cumulative Gain (NDCG):

$$NDCG@K = \frac{DCG@K}{\max DCG@K} \tag{3}$$

$$DCG@K = \sum_{i=1}^{K}\frac{2^{l_i}-1}{\log(1+i)} \tag{4}$$

Where Discounted Cumulative Gain (DCG) *K* represents the first *K* results, and l_i is taken as the correlation label of the *i* document, which is derived in four levels (0,1,2,3) from Table 2.

4. Results and discussions

4.1. SAO and FOP comparison results

As exemplified in Table 5, preliminary results can illustrate that the most of SAO triples mainly center on verbs or verb phrases, which disable them to focus on the non-functional and non-triple relations, which easily neglect some vital sentence structures in the contents. In contrast, the FOP combinations derived from promising extraction methods are suitable for the semantic understanding of patent contents with grain-refined functional labels and elaborate functional relationships. In the text1, the feature details are derived in the adjacent FOP formats of " (second device, couple,) and (couple, first device,)". As for the knowledge representations, the Glove pre-trained language models (LM) of patent claims are adopted for the downstream tasks, as well as the domain-specific similarity calculations (Pennington et al., 2014). In Table 6, one simple but necessary embedding application of the Glove vectors pre-trained from a large-scale patent corpus is utilized as a visualization result of the average embeddings of FOP structures, which means the vector-space capturing of textual concepts or phrases in patent applications all over technology areas. Additionally, the corresponding SAO embeddings, keywords, IPC codes, etc., are also prepared and loaded for the subsequent experiments.

Table 5
Comparison of SAO and FOP extraction results.

Text1	SAO triples	FOP combinations
a second device electrically coupled to the first device operable to receive the working list and being operable to program a camera view location, the second device comprising: a second memory for storing at least one working list of characters	(second device, receive, working list) (second device, comprise, second memory)	(second device, couple,) (couple, first device,) (receive, list, operable) (program, camera view,) (store, second memory, at least) (work, character,)

Table 6
The FOP knowledge representations with average 300-dimensional vectors.

Type	Term	V_1	V_2	V_3	.....	V_{298}	V_{299}	V_{300}
Function	correspond	0.0226	0.0085	0.3298	.....	0.1267	0.0682	0.1831
Object	second memory	0.1441	0.2975	0.4147	.....	0.8642	0.5861	0.4326
Property	programmed	0.0972	0.1511	0.2470	.....	0.0236	0.1636	0.3287
Functional Semantic Knowledge (FOP)	(correspond, second memory, programmed)	0.0880	0.1524	0.3303		0.3382	0.2726	0.3148

Table 7
Direct matching results using the matching dataset with SAO and FOP combinations.

Category	Lexical-based			Vector-based				Knowledge-based		
Method	Dice	Inclusion	Jaccard	Euclidean	Pearson	Spearman	Cosine	Lin	Resnik	Jiang
SAO	0.3748	0.4165	0.3575	0.2706	0.4868	0.2857	**0.4852**	0.3197	0.3263	0.1102
FOP	**0.4748**	**0.4764**	**0.4681**	**0.2735**	**0.4896**	**0.2881**	**0.4852**	**0.3943**	**0.4142**	**0.2228**

Table 8
Ranking results using the ranking dataset with SAO and FOP combinations.

Category	Method	Dice	Inclusion	Jaccard	Euclidean	Pearson	Spearman	Cosine	Lin	Resnik	Jiang
Top N@100	SAO	24.0	23.3	24.0	13.8	6.0	6.8	4.9	9.0	9.5	8.8
	FOP	**10.9**	**10.8**	**12.1**	**5.8**	**4.3**	**4.3**	**4.9**	**5.7**	**5.6**	**5.2**
MAP	SAO	0.0644	0.0835	0.0598	0.1525	0.1865	0.1803	0.2275	0.3148	0.3750	0.1865
	FOP	**0.1111**	**0.1083**	**0.1174**	**0.5982**	**0.5774**	**0.6203**	**0.4236**	**0.5357**	**0.4732**	**0.1977**
NDCG	SAO	0.2449	0.2694	0.2395	0.3407	0.3734	0.3655	0.3960	0.4773	0.5258	0.3734
	FOP	**0.2991**	**0.2961**	**0.3057**	**0.6910**	**0.6724**	**0.6934**	**0.5505**	**0.6488**	**0.5987**	**0.3965**

The first patent STS dataset (i.e., the patent matching dataset) is herein applied to complement the comparative experiments. As shown in Table 7, the direct similarity calculation results using SAO and FOP features in different methods, are obtained and collected. Without any weighted strategies or other tricks, the preliminary results have demonstrated that FOP structures with more semantic details intentionally added are explicitly emerging more advantageous than SAO triples no matter using any similarity algorithm in a statistical manner. To be further, apart from the direct comparisons, the ranking dataset is also employed as one another criteria for judging the performances of similarity approaches in the simulated patent retrieval. As shown in Table 8, the ranking results using FOP structures still seem more helpful than those of SAO triples in adequately identifying objective patents at higher ranking levels. It might be inferred that the FOP features can take all-round functional semantic knowledge into account if incorporated with technical attributes and parameters in various categories. These results are identical with the nature of functional operations of patented inventions. With the transcribing supplement of hierarchical forms such as FO, FP, OP, OFP, FOP and individual phrases of patent contents, most of the semantic deviations would seem probable to be automatically complemented.

4.2. Direct similarity calculation results

4.2.1. Comparison analysis with IPC classification

The patent search in the current stage often relies on the IPC codes, related keywords and database constraints. With the FOP combinations taken into account, the keyword features can be replaced in the semantic evaluation. In contrast, the IPC codes have to be

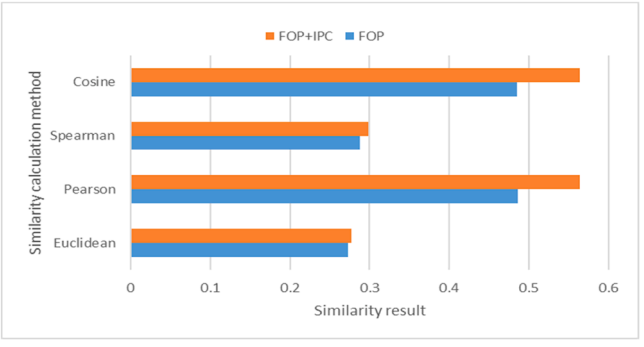


Fig. 9. Comparative similarity calculation results using "FOP" and "FOP + IPC" features.

reserved because the specific technological field can be coarsely located using the universal patent classification. The experiments should be conducted to analyze the IPC's performance and verify its effectiveness by combining IPC classification codes with FOP knowledge. As shown in Fig. 9, the results have demonstrated that the introduction of IPC codes has made a significant effect on the improvement of the similarity calculation, which indicates that IPC codes should be incorporated in the patent retrieval.

4.2.2. *Feature weights' analysis*

Intuitively, the importance of each FOP structure should be quite different because of the domain-specific technological features. However, the actually heterogeneous distribution of SAO or FOP combinations are usually encountered in the practice. In order to overcome these uneven concentration issues of patent features, the weights should be non-uniformly assigned on FOP combinations. Generally speaking, the patent contents mainly consist of two parts: the normal descriptions and the key features. To be simplified, the experiment allocates the weighted ratios on the "normal and key" segments of the contents in the patent matching dataset. In fact, the normal content part contains non-credibility combinations, and contrastively key feature maps show significantly cumulative technological information related to the patent inventions. Therefore, the weighted ratios can be manually configured on two parts, so called "Normal and Key" contents, and results are shown in Fig. 10. The results obtained from different weight ratios have demonstrated that the greater the weights of the key feature maps, the better the results. This indicates that the focus on key technical features can improve the similarity results to a certain extent. Of course, the selected FOP features should be put more focus incorporated with weights' strategies, which can effectively neglect the influence of the non-uniform distributed amount of FOP structures among the patent contents. This is also one of the advantages over the SAO-based methods.

4.2.3. *Pre-trained vector computation*

With the help of FOP knowledge representations, the patent contents can gradually be elaborated upon this approach to synthesize dense vectors of words which consult the basic ideas of the Glove models (Pennington et al., 2014). The Glove's training objective is to make the scalar product of two embedded vectors equal to the logarithm expression of words' co-occurrence probability in a bulk of corpus, which is better to reflect statistical distributions of words in the corpus. As a result, the pre-trained Glove embeddings allow for words, sentences and documents to be mapped in the same space in the convenience of vector-based calculations. We herein utilize two types of training corpuses: a global corpus collected from the website and a manipulated patent corpus. The global corpus consists of varying Wikipedia dumps between the year from 2010 to 2014. In contrast, the patent corpus is composed of millions of USTPO patents with titles, abstracts and claims utilized. The two types of pre-trained word vectors are comparatively tested using vector-based approaches in the matching dataset and ranking dataset, and the results are presented from Fig. 11 (a) to (d). The direct similarity results have shown that the patent Glove embeddings are inferred to be effective in improving the performance of the similarity calculation. Moreover, the MAP, MRR, and NDCG results of the ranking dataset are also given, which provide the identical conclusions supporting that the pre-trained vectors using patent corpus would be more helpful, as discussed above. The reason could be that the pre-trained vectors from a set of patent corpus can take more technological characteristics of patented inventions into consideration, which contributes a lot to the more accurate computation of the statistical distribution of relevant technical elements and further account for the more appropriate word embeddings as the pre-trained results.

4.3. *Deep matching calculation results*

To analyze the performance of deep learning algorithms, two types of deep matching models described in the Section 3.4.2 are also designed and developed for the comparative experiments using the modified patent STS datasets. In order to meet the requirements of the actual tasks, the paired patents with a score of 2–3 is considered as a positive sample set as 1. In contrast, a negative sample is assigned 0, which is also constructed with the targeted patent and a non-feature matched patent randomly selected from the global patent database. Finally, we have mixed ten thousand of the positive and negative samples to support deep matching tasks where the numbers of the training set, validation set, and test set are divided with a ratio of "6: 2: 2", respectively. Considering the flexibility

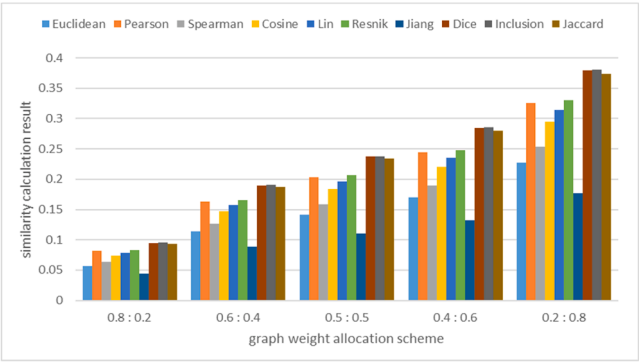
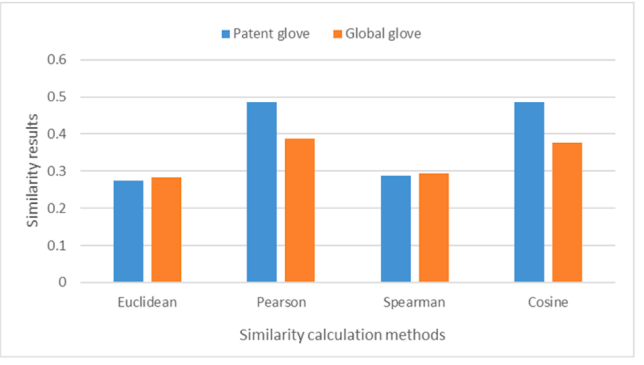
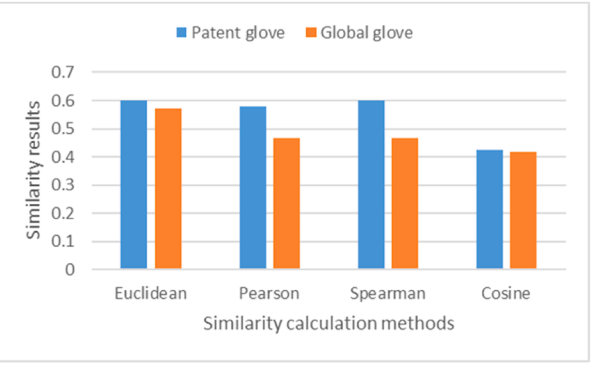


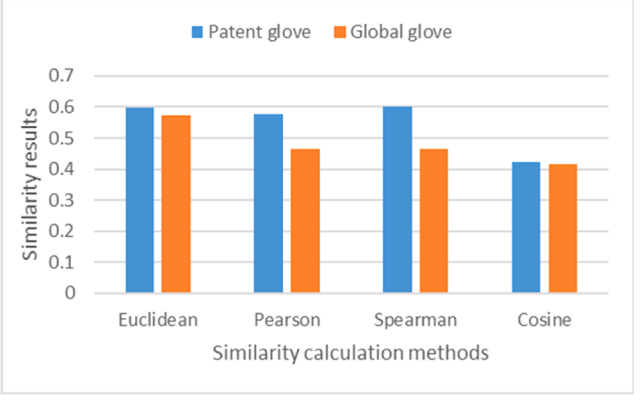
Fig. 10. Similarity results with different weight ratios assigned on"Normal: Key" patent contents.



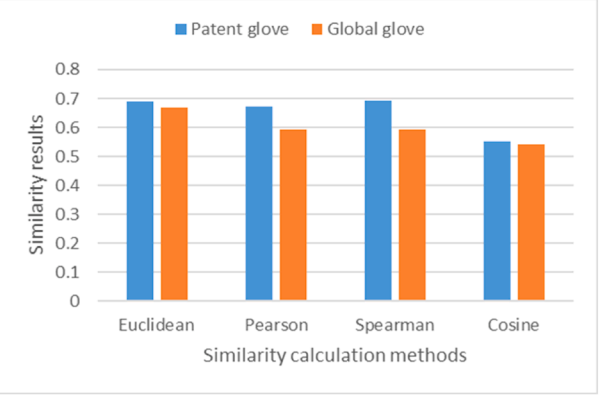
(a) Direct matching results using different Gloves



(b) MAP results using different Gloves



(c) MRR results using different Gloves



(d) NDCG results using different Gloves

Fig. 11. Similarity results using different types of Glove embeddings.

Table 9
Training parameters of the deep matching algorithms.

Parameter	Value
Computing platform	Python 3.6
Deep learning framework	Tensorflow 1.14.0
Loss	Cross-entropy
Optimzer	Adam
Batch size	64
Epoch	15
Embed size	300

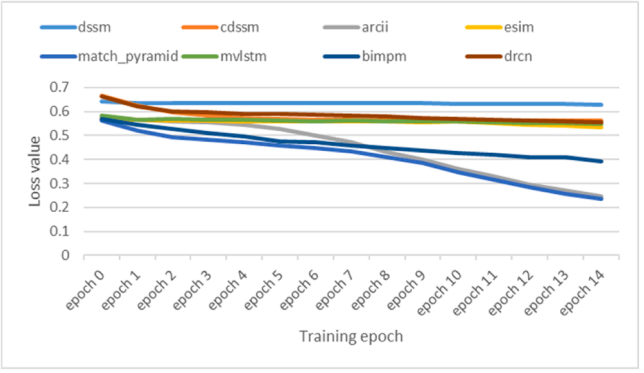


Fig. 12. Loss curves of different deep models.

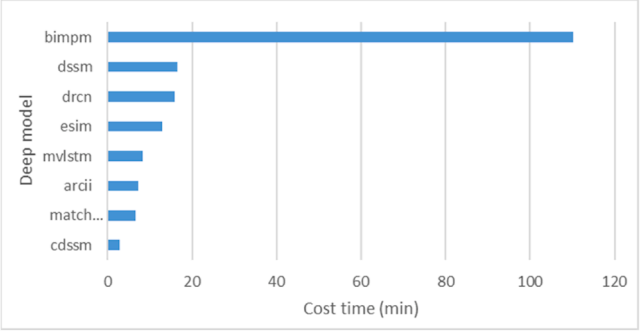


Fig. 13. Time consumptions of different deep models.

needs and the easier use, the deep learning framework is developed utilizing the Python 3.6 language and TensorFlow 1.14.0 platform. In the training processes of deep neural network models, appropriate parameters often have a great influence on the results. According to the actual situations, most of the training parameters configured in this experiment are shown in Table 9. After the training and validation steps, the training losses and models' time-consumptions in the training process are shown in Figs. 12 and 13. Obviously, the interaction-type models at the early stage, such as the BIMPM model, have a larger time-consuming cost than others. Moreover, the reasonable selection of deep matching models not only depends on the validation results, but also the relevant training efficiency, model parameters and final matching performances. Therefore, the further comparisons should be explored.

In this experiment, the matching and ranking results obtained from deep learning models mentioned above are shown in Table 10. Although the Siamese-type models (such as the DSSM) sometimes have higher vectorized similarity results, they cannot quite present another excellent performance in the ranking results. In contrast, the interaction-based models (such as the ARCII, DRCN and ESIM models) have been depicted with the generally superior performances in the similarity or ranking calculation, which is inferred with the fact that plentiful interactions not only encode finer-grained phrase representations but also grasp the details of the combined FOP knowledge interactions. Through the deep representation learning of local functional semantic knowledge, the semantic relationships can be more deeply mined and the semantic similarity results can be promoted.

Table 10
Matching and ranking results using different similarity calculation methods.

Category	Model	Euclidean		Pearson		Spearman		Cosine	
		Sim	Rank	Sim	Rank	Sim	Rank	Sim	Rank
Direct	—	.2735	5.8	.4896	4.3	.2881	4.3	.4852	4.9
Siamese-based	DSSM	.3365	57	**.6601**	61	**.7139**	64	.7310	61
	CDSSM	.4301	24	.2451	57	.5982	62	.3190	25
Interaction-based	ARCII	.2516	**22**	**.7567**	**19**	**.6891**	**24**	**.7749**	**19**
	MVLSTM	.1494	66	.1709	66	.1663	68	.3648	66
	DRCN	**.6828**	23	.2704	48	.3325	33	**.8019**	38
	BIMPM	.4515	60	.1799	**36**	.2976	45	.3254	34
	ESIM-AVG	.4043	**14**	.5658	62	.6317	49	.4722	62
	ESIM-MAX	**.6434**	38	.2538	50	.2974	**31**	.5274	**10**
	MATCH PYRAMID	.2789	51	.5954	50	.5948	50	.6190	50

4.4. Discussions and remarks

According to the descriptions of each model introduced in Section 3.4.2, the attention mechanism as a weight allocation scheme, is successfully utilized to process more interaction features and achieve local and global referenced mappings despite limited computing resources. As a comprehensive view, the interaction-type deep models should be selected with the computation efficiency and domain-specific deployments consideration. Taking the phrase order, entity type and relations, the semantic vector generation of FOP combinations can be divided into two types: the spliced vector of FOP elements' vectors and the averaged vector of each FOP element's vector summation. As shown in Fig. 14, the direct similarity results have demonstrated that the averaged-type embeddings seem to be better than the spliced-type ones. This might be inferred by the fact that the denser vector representations of FOP combinations could provide more support to the feature learning in a universal semantic space.

Up until now, an optimally selected similarity calculation frameworks can be summarized with the key parameters verified, as shown in Table 11. In this framework, the fine-grained types of patent knowledge should be extracted and flexibly constituted, such as the FOP combinations. Additionally, the word extension tools and IPC codes should also be adopted. More weights are also suggested to put on the key features selected from patent segments. One of the interaction-type deep matching models can be utilized with the averaged patent pre-trained vectors of FOP combinations. In the next, one of similarity methods following this framework will be actually validated in case studies.

5. Case studies

To further evaluate the performance of our methodology using the functional semantic knowledge (FOP), we have selected three patents with representative features, including different titles, IPC codes, competitive applicants, multiple databases, involved in litigation or infringement affairs from our patent STS datasets. The basic situation of three targeted patents is described as follows: the IBM patent (ID: US20050125556A1, IPC: G06F9/50) proposed a system and a method of managing data movement, in which a processing environment was established in a cluster of nodes. In contrast, the patent US20080130433A1 (IPC: G11B7/00) described a drive apparatus of the present invention including a recording/reproduction section, a drive control section and a memory circuit. The Motorola patent US20060099963A1 (IPC: H04Q7/20) had proposed a method of providing the location-based services using a wireless communication system that facilitated the communication with a plurality of communication units. In contrast, the other three objective patents are listed here. The Silicon Graphics (SGI) patent with the ID number of US20040249904A1 (IPC: G06F15/167), which is the objective patent of US20050125556A1, had proposed a cluster of computer system nodes connected by a storage area

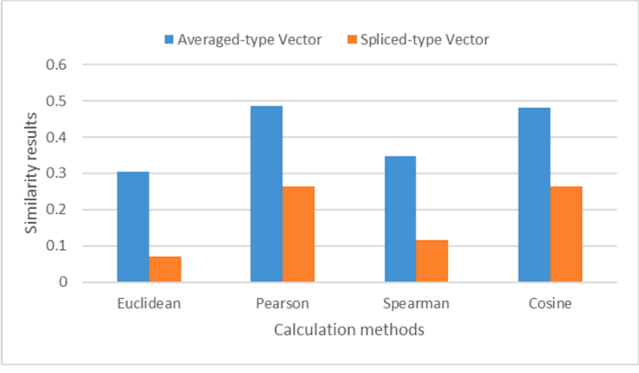


Fig. 14. Direct similarity results using different types of embedded vectors.

Table 11
New similarity calculation framework with the optimally selected parameters.

Category	Result	FOP with deep learning
Extraction strategy	Element extraction	NER (Bi-LSTM+CRF)
	Triple element	Function, Object, Property
	Contained type	FO, FP, OP, FOP, OFOP, etc.
	Extend tool	WordNet
Calculation strategy	Input content	FOP&IPC
	Text weight	"Key: Normal" ratio = 0.8: 0.2
	Vector type	Averaged deep representation vector
	Deep Matching	Max{ARCII, DRCN, ESIM}

network including two classes of nodes. US20040151090A1 (IPC: G11B7/05, Priority-Country is Japan), which is the objective patent of US20080130433A1, gave a data recording/reproducing method and apparatus. In a disk drive device, it was also described as a specific embodiment of the invention, and IP cluster addresses were associated with an address unit. US20010018349A1 (H04Q7/20, Priority-Country is Finland) is the objective patent of US20060099963A1, and it helped describe a system for providing location dependent services to a plurality of mobile terminals within a coverage area. These three paired patents are located in their corresponding sets of patents in the ranking dataset as described in the Section 4.1. The objective patents should be searched to be ranked as high positions as possible in the retrieval. Generally, the patent retrieval is mainly conducted using the global patent database. We herein carried out the search processes using our ranking dataset to simulate the actual retrieval process as much as possible.

In this case, three similarity methods are used in the simulated search. As shown in Tables 12 and 13, the recalled ranking results using a knowledge-based similarity method in the comparison of SAO and FOP structures have preliminarily demonstrated the FOP-based method can better identify objective patents with relatively higher topN@100 ranks (the bold patents are the ones that were recorded as the referenced objective ones). In order to further assess the effectiveness of FOP-based methods, some representatives of the interaction-based models, such as the ARCII, ESIM and DRCN models, are preferred with involvements in another experiment of the simulated patent retrieval. As shown in Table 14, the results have depicted that deep models should improve the retrieval ranking results with averaged FOP representations embedded in deep models. The interaction-type models (such as ESIM) can make the appealing patents ranked higher, which are in accordance with the bi-directional sequence encoding and the attention enhancement of the interactive semantic learning from FOP elements. Integrated with the local inferenced semantics, the model could grasp the more accurately semantic commonality between patented pairs. As is summarized in Table 15, the comparison results using three methods have explicitly presented that the FOP-based similarity calculation framework should be incorporated with effective extraction methods (i.e., the NER identification), flexible combinations of FOP knowledge related to the technical segments, higher weights assigned on the feature maps, averaged patent pre-trained embeddings and knowledge-based similarity algorithms. In this process, deep matching models, especially the interaction-type models should be implemented to further obtain a reasonable matching result. In the near future, the FOP knowledge combinations with their WordNet expansions would be further validated using the deep models in the similarity calculation.

6. Conclusions

In order to discover and mine the similar technological intelligence in the patents, this paper proposes a kind of functional semantic knowledge (FOP) which helps to enhance the patent semantic text similarity. Finally, an entire similarity framework is reasonably constructed through a series of systematic analysis and comparative experiments with the FOP-based approaches. Now the conclusions can be summarized as follows.

1) A kind of functional semantic knowledge (FOP) from patents has firstly been proposed instead of SAO structures for the computational similarity. In order to reasonably evaluate the similarity methods, new patent semantic text similarity (STS) datasets

Table 12
Similarity calculation results using SAO features.

US20050125556A1			US20080130433A1			US20060099963A1		
Patent ID	Similarity	Rank	Patent ID	Similarity	Rank	Patent ID	Similarity	Rank
US4857111A	0.512	1	US07055737B1	0.519	1	US06966060B1	0.445	1
EP0778023A1	0.468	2	US20020181435A1	0.517	2	**US20010018349A1**	**0.440**	**2**
US7098200B2	0.451	3	US20050205543A1	0.488	3	US20020035380A1	0.399	3
US20030023336A1	0.430	4	US20040122488A1	0.487	4	US20030195571A1	0.385	4
US20040037279A1	0.401	5	US20040268154A1	0.487	5	US20040024588A1	0.378	5
US20030210710A1	0.390	6	US20060195816A1	0.482	6	US20030189922A1	0.377	6
US20030160826A1	0.386	7	US20060031283A1	0.469	7	US20060140406A1	0.376	7
US20030129576A1	0.386	8	US20030234627A1	0.463	8	US20040126030A1	0.373	8
US20040179689A1	0.383	9	**US20040151090A1**	**0.455**	**9**	US20030153953A1	0.372	9
US20040249904A1	**0.383**	**10**	US20020171871A1	0.445	10	US20020024536A1	0.360	10

Table 13
Similarity calculation results using FOP combinations.

US20050125556A1			US20080130433A1			US20060099963A1		
Patent ID	Similarity	Rank	Patent ID	Similarity	Rank	Patent ID	Similarity	Rank
US20030004936A1	0.469	1	US20070232896A1	0.461	1	US20060140406A1	0.557	1
US20020124205A1	0.468	2	US20020181435A1	0.459	2	**US20010018349A1**	**0.511**	**2**
US20030210710A1	0.463	3	**US20040151090A1**	**0.457**	**3**	US20030189922A1	0.504	3
US20010048342A1	0.462	4	US20020157515A1	0.456	4	US20050085865A1	0.495	4
US20040249904A1	**0.458**	**5**	US20030227651A1	0.456	5	US20040028129A1	0.491	5
US20040179689A1	0.456	6	US20040122488A1	0.456	6	US20020024536A1	0.488	6
US20030187854A1	0.454	7	US20040185857A1	0.450	7	US20020025427A1	0.471	7
US5077486A	0.451	8	US20030234627A1	0.445	8	US20040024588A1	0.466	8
US20030129576A1	0.448	9	US20050268764A1	0.442	9	US06966060B1	0.451	9
US20030023336A1	0.436	10	US20020171871A1	0.436	10	US20030153953A1	0.442	10

Table 14
Similarity calculation results using FOP combinations and interaction-based deep models.

US20050125556A1			US20080130433A1			US20060099963A1		
Patent ID	Similarity	Rank	Patent ID	Similarity	Rank	Patent ID	Similarity	Rank
US20020149154A1	0.661	1	US20020106689A1	0.624	1	**US20010018349A1**	**0.656**	**1**
US20040249904A1	**0.625**	**2**	US20020181435A1	0.586	2	US20040028129A1	0.567	2
US20030120825A1	0.564	3	**US20040151090A1**	**0.520**	**3**	US20020035380A1	0.504	3
US20020142992A1	0.542	4	US20020141479A1	0.494	4	US20030153953A1	0.493	4
US20030160826A1	0.529	5	US20060242466A1	0.482	5	US20030195571A1	0.493	5
US20030188184A1	0.514	6	US20060218114A1	0.473	6	US20020024536A1	0.482	6
US20040034370A1	0.504	7	US20040122488A1	0.466	7	US20030189922A1	0.436	7
US20020023190A1	0.502	8	US20060131293A1	0.465	8	US20040126030A1	0.429	8
US20030104341A1	0.501	9	US20030195780A1	0.453	9	US20040138719A1	0.427	9
US4857111A	0.495	10	US20030227651A1	0.439	10	US20040029041A1	0.418	10

Table 15
Summarized similarity calculation results of three different methods.

Category	Result	SAO	FOP	FOP with deep learning
Rank	Rank for patent 1	10	5	**2**
	Rank for patent 2	9	3	**3**
	Rank for patent 3	2	2	**1**
Average	Rank result	7.0	3.3	**2.0**
	Similarity	0.426	0.475	**0.600**
	Computation time (mins)	1.682	0.353	**0.005**
Extraction	Extraction	POS-based	Bi-LSTM + CRF	Bi-LSTM + CRF
	Element	Function, Object	Function, Object, Property	Function, Object, Property
	Contained type	SA, AO, SAO	FO, FP, OP, FOP, OFP	FO, FP, OP, FOP, OFP
	Extend tool	—	WordNet	WordNet
Calculation	Input	SAO	FOP	FOP
	Vector type	Global Vectors	Patent Vectors	Averaged Patent Vectors
	Deep Matching	—	—	Max {ARCII, DRCN, ESIM}

including the matching dataset and the ranking dataset, are well prepared and released later as an open-source project (see more details in https://github.com/openKG-field).

2) Direct similarity results using lexical-based, vector-based and knowledge-based calculation methods have preliminarily demonstrated that FOP-based methods are more appropriate than SAO-based ones in the semantic matching. If the universal classification references (such as IPC codes) combined, the similarity results and even the actual patent retrieval would be further improved.
3) Key factors including weights' assignments and pre-trained vectors are also discussed in details using patent STS datasets. Key feature maps of patents should be deployed with higher weight ratios, and pre-trained embeddings based on patent corpus deserve more concerns in the semantic text similarity.
4) The deep matching models have also been fully investigated, and experiment results show that most of the interaction-based models can effectively improve the semantic learning capability, and meanwhile reduce the complexity of feature combination computation with a higher "cost-performance" ratio. The averaged pre-trained embeddings of FOP combinations are much more suggested to enhance the matching efficiency.
5) A new patent similarity calculation framework is summarized incorporated with various aspects including the FOP extraction, combinations, knowledge extensions, weights' assignments, pre-trained patent embeddings and interaction-based models. In the

case studies, one of the optimally selected methods has made a marvelous contribution to allocate higher ranking positions of objective patents in the simulated patent retrieval.

Of course, some problems still exist: more accurate SAO and FOP extraction methods should be introduced in favor of the comprehensive comparison. Additionally, the influential factors of interaction-type deep models would be further evaluated by ablation studies. In the simulated retrieval, the number of the ranking dataset is relatively small and the retrieval efficiency waits to be further explored. In the future, we will put more focus on the large-scale pre-trained language models (such as BERT or GPT) in advancing the sentence-level understanding of semantic tasks. On the other hand, cutting-edge deep graph networks and complicated matching architectures should also be explored with representation learning algorithms developed. More reasonable approaches using the FOP structures of patents for enhancing the semantic text similarity would be addressed.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abbas, A., Zhang, L., & Khan, S. U.. (2014). A literature review on the state-of-the-art in patent analysis. *World Patent Information, 37*, 3–13. https://doi.org/10.1016/j.wpi.2013.12.006

Agirre, E., Gonzalez-Agirre, A., Lopez-Gazpio, I., Maritxalar, M., Rigau, G., & Uria, L. (2016). SemEval-2016 task 2: Interpretable semantic textual similarity. In *SemEval 2016 - 10th International Workshop on Semantic Evaluation, Proceedings* (pp. 512–524). https://doi.org/10.18653/v1/s16-1082

Amir, S., Tanasescu, A., & Zighed, D. A. (2017). Sentence similarity based on semantic kernels for intelligent text retrieval. *Journal of Intelligent Information Systems, 48*(3), 675–689. https://doi.org/10.1007/s10844-016-0434-3

An, X., Li, J., Xu, S., Chen, L., & Sun, W. (2021). An improved patent similarity measurement based on entities and semantic relations. *Journal of Informetrics, 15*(2), Article 101135. https://doi.org/10.1016/j.joi.2021.101135

Arts, S., Cassiman, B., & Gomez, J. C. (2018). Text matching to measure patent similarity. *Strategic Management Journal, 39*(1), 62–84. https://doi.org/10.1002/smj.2699

Cascini, G., & Zini, M. (2008). Measuring patent similarity by comparing inventions functional trees. In G. Cascini (Ed.), *IFIP international federation for information processing (Vol. 277, pp. 31–42)*. Springer US. https://doi.org/10.1007/978-0-387-09697-1_3.

Chen, L., Xu, S., Zhu, L., Zhang, J., Lei, X., Yang, G., & Chen, L. (2020). A deep learning based method for extracting semantic information from patent documents. *Scientometrics, 125*(1), 289–312. https://doi.org/10.1007/s11192-020-03634-y

Choi, J., & Hwang, Y. S.. (2014). Patent keyword network analysis for improving technology development efficiency. *Technological Forecasting and Social Change, 83*(1), 170–182. https://doi.org/10.1016/j.techfore.2013.07.004

Choi, S., Yoon, J., Kim, K., Lee, J. Y., & Kim, C. H. (2011). SAO network analysis of patents for technology trends identification: A case study of polymer electrolyte membrane technology in proton exchange membrane fuel cells. *Scientometrics, 88*, 863–883. https://doi.org/10.1007/s11192-011-0420-z

Cong, H., & Tong, L. H.. (2008). Grouping of TRIZ Inventive Principles to facilitate automatic patent classification. *Expert Systems with Applications, 34*(1), 788–795. https://doi.org/10.1016/j.eswa.2006.10.015

Dagan, I., Glickman, O., & Magnini, B. (2006). The PASCAL recognising textual entailment challenge. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 3944*, 177–190. https://doi.org/10.1007/11736790_9. LNAI (May 2014).

Das, D., Chen, D., Martins, A. F. T., Schneider, N., & Smith, N. A. (2014). Frame-semantic parsing. *Computational Linguistics, 40*(1), 9–56. https://doi.org/10.1162/COLI_a_00163

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, 1*, 4171–4186. http://arxiv.org/abs/1810.04805.

Feng, L., Niu, Y., Liu, Z., Wang, J., & Zhang, K. (2020). Discovering technology opportunity by keyword-based patent analysis: A hybrid approach of morphology analysis and USIT. *Sustainability (Switzerland), 12*(1), 1–35. https://doi.org/10.3390/SU12010136

Fiorineschi, L., Frillici, F. S., & Rotini, F. (2020). Enhancing functional decomposition and morphology with TRIZ: Literature review. *Computers in Industry, 115*(January), 1–15. https://doi.org/10.1016/j.cad.2011.12.006

Giabelli, A., Malandri, L., Mercorio, F., Mezzanzanica, M., & Nobani, N. (2022). Embeddings evaluation using a novel measure of semantic similarity. *Cognitive Computation, 14*, 749–763. https://doi.org/10.1007/s12559-021-09987-7. July 2021.

Hain, D. S., Jurowetzki, R., Buchmann, T., & Wolf, P. (2022). A text-embedding-based approach to measuring patent-to-patent technological similarity. *Technological Forecasting and Social Change, 177*, Article 121559. https://doi.org/10.1016/j.techfore.2022.121559

He, H., & Lin, J. (2016). Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference* (pp. 937–948). https://doi.org/10.18653/v1/n16-1108

He, X., Meng, X., Wu, Y., Chan, C. S., & Pang, T. (2020). Semantic matching efficiency of supply and demand texts on online technology trading platforms: Taking the electronic information of three platforms as an example. *Information Processing and Management, 57*(5), Article 102258. https://doi.org/10.1016/j.ipm.2020.102258

Hu, B., Lu, Z., Li, H., & Chen, Q. (2014). Convolutional neural network architectures for matching natural language sentences. *Advances in Neural Information Processing Systems, 3*(January), 2042–2050.

Huang, P. Sen, He, X., Gao, J., Deng, L., Acero, A., & Heck, L. (2013). Learning deep structured semantic models for web search using clickthrough data. In *International Conference on Information and Knowledge Management, Proceedings*. https://doi.org/10.1145/2505515.2505665

Hussain, M. J., Wasti, S. H., Huang, G., Wei, L., Jiang, Y., & Tang, Y. (2020). An approach for measuring semantic similarity between Wikipedia concepts using multiple inheritances. *Information Processing and Management, 57*(3), Article 102188. https://doi.org/10.1016/j.ipm.2019.102188

Inan, E. (2020). SimiT: A text similarity method using lexicon and dependency representations. *New Generation Computing, 38*(3), 509–530. https://doi.org/10.1007/s00354-020-00099-8

Jang, M.-H., Eom, T.-H., Kim, S.-W., & Hwang, Y.-S. (2016). Document similarity measure based on the earth Mover\'s distance utilizing latent dirichlet allocation. *Research Journal of Applied Sciences, Engineering and Technology, 12*(2), 214–222. https://doi.org/10.19026/rjaset.12.2323

Ji, S., Pan, S., Cambria, E., Marttinen, P., & Yu, P. S. (2022). A survey on knowledge graphs: representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems, 33*(2), 494–514. https://doi.org/10.1109/TNNLS.2021.3070843

Kohila, R. (2016). K. A.text mining: text similarity measure for news articles based on string based approach. *Global Journal of Engineering Science and Research Management, 3*(7), 35–42.

Krestel, R., Chikkamath, R., Hewel, C., & Risch, J. (2021). A survey on deep learning for patent analysis. *World Patent Information, 65*, Article 102035. https://doi.org/10.1016/j.wpi.2021.102035

Kwon, S., Oh, D., & Ko, Y. (2021). Word sense disambiguation based on context selection using knowledge-based word similarity. *Information Processing and Management, 58*(4), Article 102551. https://doi.org/10.1016/j.ipm.2021.102551

Guarino, G., Samet, A., & Cavallucci, D. (2022). PaTRIZ: A framework for mining TRIZ contradictions in patents. *Expert Systems with Applications, 207*, 117942. https://doi.org/10.1016/j.eswa.2022.117942

Lan, W., & Xu, W. (2018). Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. ArXiv. http://arxiv.org/abs/1806.04330.

Lee, C., Song, B., & Park, Y. (2013). How to assess patent infringement risks: A semantic patent claim analysis using dependency relationships. *Technology Analysis and Strategic Management, 25*(1), 23–38. https://doi.org/10.1080/09537325.2012.748893

Leydesdorff, L., Kushnir, D., & Rafols, I. (2012). Interactive overlay maps for US patent (USPTO) data based on International Patent Classification (IPC). *Scientometrics, 98*(3), 1583–1599.

Li, X., Wang, C., Zhang, X., & Sun, W. (2020). Generic SAO similarity measure via extended sorensen-dice index. *IEEE Access : Practical Innovations, Open Solutions, 8*, 66538–66552. https://doi.org/10.1109/ACCESS.2020.2984024

Liu, X. (2013). Full-text citation analysis : A new method to enhance. *Journal of the American Society for Information Science and Technology, 64*(July), 1852–1863. https://doi.org/10.1002/asi

Lyu, B., Chen, L., Zhu, S., & Yu, K. (2021). LET: Linguistic knowledge enhanced graph transformer for chinese short text matching. In *Thirty-Fifth AAAI Conference On Artificial Intelligence, Thirty-Third Conference On Innovative Applications Of Artificial Intelligence And The Eleventh Symposium On Educational Advances In Artificial Intelligence* (pp. 13498–13506). *35* http://arxiv.org/abs/2102.12671.

Majumder, G., Pakray, P., Gelbukh, A., & Pinto, D. (2016). Semantic textual similarity methods, tools, and applications: A survey. *Computacion y Sistemas, 20*(4), 647–665. https://doi.org/10.13053/CyS-20-4-2506

Meek, W.Y.C. (2018). W IKI QA : A challenge dataset for open-domain question answering. September 2015, 2013–2018. http://www.aclweb.org/anthology/D15-1237.

Manning, C. D., Bauer, J., Finkel, J., & Bethard, S. J. (2014). *The stanford CoreNLP natural language processing toolkit* (pp. 55–60). Baltimore, Maryland: Association for Computational Linguistic. http://macopolo.cn/mkpl/products.asp.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2019). Advances in pre-training distributed word representations. In *LREC 2018 - 11th International Conference on Language Resources and Evaluation* (pp. 52–55).

Miller, G. A. (1995). WordNet: A lexical database for english. *Communications of the ACM, 38*(11), 39–41. https://doi.org/10.1145/219717.219748

Park, H., Kim, K., Choi, S., & Yoon, J. (2013a). A patent intelligence system for strategic technology planning. *Expert Systems with Applications, 40*(7), 2373–2390. https://doi.org/10.1016/j.eswa.2012.10.073

Park, H., Ree, J. J., & Kim, K. (2012). An SAO-based approach to patent evaluation using TRIZ evolution trends. In *2012 IEEE 6th International Conference on Management of Innovation and Technology ICMIT 2012*. https://doi.org/10.1109/ICMIT.2012.6225873

Park, H., Ree, J. J., & Kim, K. (2013b). Identification of promising patents for technology transfers using TRIZ evolution trends. *Expert Systems with Applications, 40*(2), 736–743. https://doi.org/10.1016/j.eswa.2012.08.008

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference* (pp. 1532–1543). https://doi.org/10.3115/v1/d14-1162

Prakoso, D. W., Abdi, A., & Amrit, C. (2021). Short text similarity measurement methods: A review. *Soft Computing, 25*(6), 4699–4723. https://doi.org/10.1007/s00500-020-05479-2

Quan, X., Liu, G., Lu, Z., Ni, X., & Wenyin, L. (2010). Short text similarity based on probabilistic topics. *Knowledge and Information Systems, 25*(3), 473–491. https://doi.org/10.1007/s10115-009-0250-y

Quan, Z., Wang, Z. J., Le, Y., Yao, B., Li, K., & Yin, J. (2019). An efficient framework for sentence similarity modeling. *IEEE/ACM Transactions on Audio Speech and Language Processing, 27*(4), 853–865. https://doi.org/10.1109/TASLP.2019.2899494

Raj, M., Tiwari, P., & Gupta, P. (2022). Cosine similarity, distance and entropy measures for fuzzy soft matrices. *International Journal of Information Technology, 14*, 2219–2230. https://doi.org/10.1007/s41870-021-00799-4

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuad: 100,000+ questions for machine comprehension of text. In *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings, June* (pp. 2383–2392). https://doi.org/10.18653/v1/d16-1264

Rodriguez, A., Kim, B., Turkoz, M., Lee, J. M., Coh, B. Y., & Jeong, M. K. (2015). New multi-stage similarity measure for calculation of pairwise patent similarity in a patent citation network. *Scientometrics, 103*(2), 565–581. https://doi.org/10.1007/s11192-015-1531-8

Saríc, F., Glavaš, G., Karan, M., Šnajder, J., & Bašíc, B. D. (2012). TakeLab: Systems for measuring semantic text similarity. **SEM 2012 - 1st Joint Conference on Lexical and Computational Semantics, 2*, 441–448.

Shen, Y., He, X., Gao, J., Deng, L., & Mesnil, G. (2014). Learning semantic representations using convolutional neural networks for web search. In *WWW 2014 Companion - Proceedings of the 23rd International Conference on World Wide Web* (pp. 373–374). https://doi.org/10.1145/2567948.2577348

Shih, M. J., Liu, D. R., & Hsu, M. L. (2010). Discovering competitive intelligence by mining changes in patent trends. *Expert Systems with Applications, 37*(4), 2882–2890.

Spreafico, C., & Russo, D. (2016). TRIZ industrial case studies: A critical survey. *Procedia CIRP, 39*, 51–56. https://doi.org/10.1016/j.procir.2016.01.165

Sun, Y., Liu, W., Cao, G., Peng, Q., Gu, J., & Fu, J. (2022). Effective design knowledge abstraction from Chinese patents based on a meta-model of the patent design knowledge graph. *Computers in Industry, 142*, Article 103749. https://doi.org/10.1016/j.compind.2022.103749

Suzgun, M., Melas-Kyriazi, L., Sarkar, S.K.,. Kominers, S.D.,. & Shieber, S.M. (.2022). The Harvard USPTO patent dataset: A large-scale, well-structured, and multi-purpose corpus of patent applications. Argiv, Ml, 1–38. http://arxiv.org/abs/2207.04043.

Viji, D., & Revathy, S. (2022). A hybrid approach of Weighted Fine-Tuned BERT extraction with deep Siamese Bi – LSTM model for semantic text similarity identification. *Multimedia Tools and Applications, 81*(5), 6131–6157, 0123456789.

Wang, M., Lu, Z., Li, H., & Liu, Q. (2015). Syntax-based deep matching of short texts. In *IJCAI International Joint Conference on Artificial Intelligence, 2015-Janua* (pp. 1354–1361).

Wang, W. M.., & Cheung, C. F.. (2011). A semantic-based intellectual property management system (SIPMS) for supporting patent analysis. *Engineering Applications of Artificial Intelligence, 24*(8), 1510–1520. https://doi.org/10.1016/j.engappai.2011.05.009

Wang, X., Ma, P., Huang, Y., Guo, J., Zhu, D., Porter, A. L., & Wang, Z. (2017). Combining SAO semantic analysis and morphology analysis to identify technology opportunities. *Scientometrics, 111*(1), 3–24. https://doi.org/10.1007/s11192-017-2260-y

Wang, X., Ren, H., Chen, Y., Liu, Y., Qiao, Y., & Huang, Y. (2019). Measuring patent similarity with SAO semantic analysis. *Scientometrics, 121*(1), 1–23. https://doi.org/10.1007/s11192-019-03191-z

Wang, Z., & Liu, Y. (2022). SEA-PS: Semantic embedding with attention to measuring patent similarity by leveraging various text fields. *Journal of Information Science*. https://doi.org/10.1177/01655515221106651, 01655515221106651.

Whalen, R., Lungeanu, A., DeChurch, L., & Contractor, N. (2020). Patent similarity data and innovation metrics. *Journal of Empirical Legal Studies, 17*(3), 615–639. https://doi.org/10.1111/jels.12261

Xu, C., Wang, H., Wu, S., & Lin, Z. (2021). Tag-enhanced dynamic compositional neural network over arbitrary tree structure for sentence representation[Formula presented]. *Expert Systems with Applications, 181*(May), Article 115182. https://doi.org/10.1016/j.eswa.2021.115182

Yang, C., Zhu, D., & Wang, X. (2017a). SAO semantic information identification for text mining. *International Journal of Computational Intelligence Systems, 10*(1), 593–604. https://doi.org/10.2991/ijcis.2017.10.1.40

Yang, C., Zhu, D., Wang, X., Zhang, Y., Zhang, G., & Lu, J. (2017b). Requirement-oriented core technological components' identification based on SAO analysis. *Scientometrics, 112*(3), 1229–1248. https://doi.org/10.1007/s11192-017-2444-5

Yoon, J., & Kim, K. (2012a). An analysis of property–function based patent networks for strategic R&D planning in fast-moving industries: The case of silicon-based thin film solar cells. *Expert Systems with Applications, 39*(9), 7709–7717.

Yoon, J., & Kim, K. (2012b). Detecting signals of new technological opportunities using semantic patent analysis and outlier detection. *Scientometrics, 90*(2), 445–461. https://doi.org/10.1007/s11192-011-0543-2

Yoon, J., Park, H., & Kim, K. (2013). Identifying technological competition trends for R&D planning using dynamic patent maps: SAO-based content analysis. *Scientometrics, 94*(1), 313–331. https://doi.org/10.1007/s11192-012-0830-6

Younge, K. A., & Kuhn, J. M. (2016). Patent-to-patent similarity: A vector space model. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2709238

Yu, C., Xue, H., Jiang, Y., An, L., & Li, G. (2021). A simple and efficient text matching model based on deep interaction. *Information Processing and Management, 58*(6), 1–15. https://doi.org/10.1016/j.ipm.2021.102738

Zhang, Z., & Pun, C. M.. (2022). Learning ordinal constraint binary codes for fast similarity search. *Information Processing and Management, 59*(3), Article 102919. https://doi.org/10.1016/j.ipm.2022.102919

Zhao, S., Huang, Y., Su, C., Li, Y., & Wang, F. (2020). Interactive attention networks for semantic text matching. *20th IEEE international conference on data mining (ICDM 2020), Sorrento, Italy,* (pp. 861–870). IEEE COMPUTER SOC. https://doi.org/10.1109/ICDM50108.2020.00095