# FOPNet：A comprehensive functional semantic knowledge graph for deep technical analysis in patents

**Nan Wang a,b,\*, Ziyi Wanb, Hongyu Zhaoc,Chang Wangb, Yake Wangb**

a Beijing Key Laboratory of Internet Culture and Digital Dissemination Research, Beijing Information Science and Technology University, Beijing 100192, China

b Computer School, Beijing Information Science and Technology University, Beijing 100192, China

c Tencent Technology (Beijing) Co.Ltd, Beijing 100037, China

\*Corresponding Email: wangnan8848@126.com

## Abstract

Primary sources employed in the patent semantic analysis usually cover patent keywords, basic features, citation data, categorization codes and Subject-Action-Object (SAO) features. Nevertheless, it is unable to offer deep technical semantic understanding of patents due to the absence of clear entity relations, distinct knowledge categories, and comprehensive knowledge engineering approaches, and therefore restrict its ability to meet the increasing demands of patent semantic analysis. In order to take advantage of the in-depth technological examination of patent contents, this paper proposes a novel patent knowledge graph called FOPNet (also known as Function-Object-Property Networks). Firstly, a functional semantic ontology has been created, which serves as the basis for building concepts, entities, and interactions related to the FOPNet. Furthermore, a framework implementing the entity recognition, relation extraction, and knowledge completion algorithms has been thoroughly established by combining pre-trained models (PTMs) and deep learning techniques. The preliminary results have shown that the entity and relation types could be accurately extracted with an average F1-score over 80%, and missing FOPNet knowledge could further be inferred and improved using representation learning and association mining algorithms. Obviously, PTMs could play a significant role in the construction development of patent knowledge graphs. The FOPNet has fully been demonstrated to offer a thorough functional semantic support in the patent retrieval application, where a state-of-the-art patent matching framework comprising retrieval datasets, matching algorithms and "recall-reranking-augmenting" logics has been constructed. The performance of this semantic retrieval augmented by the larage language model (LLM) has also been validated with an average ranking of Hit@10 and detailed explanations. Therefore, the FOPNet technique described in this work has a great deal of promise for a widespread acceptance in resolving diverse patented issues.

## Keywords

FOPNet, TRIZ, Functional Semantic Analysis,Patent Knowledge Graph, Large Language Model (LLM), Patent Retrieval

## Introduction

Patent collections are a crucial source of technological knowledge for academic and industrial groups around the world (Krestel et al., 2021). Important patent concepts include the International Patent Classification (IPC) codes, the patent number, and literary components such titles, abstracts, and claims. Additionally, a cited or citation analysis and a qualitative examination of the related texts enhance patent documents' bibliographic quality(Wang and Cheung, 2011; Yang, Zhu and Wang, 2017) . Nevertheless, the indicator analysis of these patent categories occasionally ignores the technological relevance. In modern bibliometrics, citation analysis primarily examines co-citation and patent citation links, allowing citation networks to be constructed based on subject categories (Liu, 2013). However, there are still certain drawbacks, such as the fact that there is a natural delay between the formation of a technical portfolio and its official acknowledgment in citations, and the citation analysis is unable to accurately reflect the state of technological advancement. When it comes to describing the claimed technology in patents, semantic features which evolved from simple keyword-based methods to more complex methods of identifying technological components, have emerged as the mainstay of patent analysis. These techniques usually require domain expertise to accurately interpret key phrases and extract useful data (J. Choi & Hwang, 2014; Feng et al., 2020). In a range of hybrid approaches, researchers have sought to integrate the citation analysis, co-word analysis, and various quantitative indicators. However, effectively identifying core technological components, such as processes, methods, and operations, continues to be a significant challenge.

Researchers have put forth a number of function-based strategies that incorporate functional conceptual models in order to overcome semantic issues in patents. Additional experiments have been conducted to identify functionalities or attributes by using Subject-Action-Object (SAO) triplets or grammatical structures like "verb + noun" to evaluate patents. These techniques support the tracking of evolutionary trends, the identification of technological hotspots, and the recommendation of novel patents(Yoon & Kim, 2012b). Nevertheless, due to variations across technical domains and patent structures, the textual characteristics of technological solutions differ accordingly. Some studies have further investigated the TRIZ principles and patent semantic relationships, uncovering the semantic links among technological problem statements, corresponding contradiction matrix and inventive solutions (Bergeaud et al., 2017), and attempting to construct knowledge representation rules to analyze patents (S. Choi et al., 2011). Although these methods have produced useful information, the semantic gap between TRIZ concepts and patent texts has not yet been effectively bridged. The significant disparity between TRIZ solutions and the detailed patent descriptions is the root cause of this challenge. Therefore, more advanced knowledge graphs for the structured feature integration, functional feature extraction, and relevance calculation of patents should be explored in order to provide a more trustworthy analytical framework (Phyo, 2022; X. Yang et al., 2021).

At the moment, general knowledge graphs frequently fall short of patent semantic computing's particular needs due to their ordinary scopes. Existing methods for assessing patent semantics are not uniformly integrated. While more suitable ontology designs are required, the process of building knowledge graphs and their application logics have not been thoroughly investigated. This study systematically explores important topics including the ontology design, knowledge preprocessing, knowledge extraction, knowledge completion, and knowledge correction in order to address these issues and provide a generic framework for the patent knowledge graph (i.e., FOPNet) creation.

## Related Works

### Patent Semantic Analysis

Patent semantic analysis, which was initially focused on engineering and technical systems, has gradually broadened to identify technological components, such as specialized knowledge connected to structures, functions, and the connections between them(Wu et al., 2020). From 2009 to 2013, the CLEF-IP forum carried out several joint projects focused on patent retrieval, structural identification, picture classification and recognition, and the novelty search using EPO datasets (Jiang, Sarica, et al., 2022). Gradually, categorized, bibliographic, and lexical-based methodologies are becoming the three main categories into which traditional patent analysis techniques can be divided (Jiang, Hu, et al., 2022; H. Park, Kim, et al., 2013). In multidisciplinary research, manual classification schemes or empirical assessments from specialists in other domains are frequently crucial (Leydesdorff et al., 2012). According to contemporary bibliometrics, patent context indicators are also combined with citation analysis or co-citation relationships, enabling the construction of a qualitative semantic analysis of the related texts according to subject categories (Wang and Cheung, 2011; Liu, 2013; Yang *et al.*, 2017). Using cross-citations and the co-occurrence of IPC classes, a framework that streamlines extensive patent citation networks was put forth to track technological convergence using new indicators that surpass the previous research (Krestel et al., 2021; Zhu & Motohashi, 2022). Furthermore, co-word analysis has shown itself to be a useful technique for rapidly deciphering textual meanings, but has a number of drawbacks: (1) synonyms or homonyms with ambiguous meanings; (2) commonly used or high-frequency phrases that often fail to distinguish themselves from relevant issues; (3) co-occurrence of phrases without explicitly predictable connections (C. Yang, Zhu, & Wang, 2017). The aforementioned techniques have proven to be highly effective in mapping technical advancements, assessing knowledge flows, and developing patent indicators (Hain et al., 2022). Moreover, the patent semantic analysis has also evolved from lexical-based approaches to more sophisticated methods for identifying technological components, often associated with key phrases that require specialized expertise (J. Choi & Hwang, 2014; Feng et al., 2020). Through hybrid approaches, researchers have attempted to integrate co-word analysis, citations, and other quantitative indicators. However, it is still very challenging to identify the key technological components, including processes, methods, and activities. As an alternative to traditional keyword-based and indicator-based approaches, a number of academics have proposed SAO-based mining techniques and other tools for analyzing unstructured data (H. Park, Ree, et al., 2013; F. Wang & Lin, 2017; C. Yang, Zhu, Wang, et al., 2017). These techniques aim to better communicate the technological essence of the patents.

### SAO-Based Methods

SAO structures have been extracted during the past few decades to represent technical solutions in patents, which has improved the semantic understanding (Choi *et al.*, 2011; Wang *et al.*, 2017; Yang, Zhu and Wang, 2017). In most cases, a subject (S), an action (A), and an object (O) make up SAO triples. Basic information, especially from patent claims, can be appropriately conveyed after SAO structures have been effectively recognized. Product-Function-Technology (PFT) maps were also utilized by the SAO-TRM method to aid in the decision-making (H. Park, Ree, et al., 2013). The SAO similarity has been used to measure the patent semantic similarity frequently with the WordNet dictionary. A SAO-based intelligent system has been designed to compute the semantic similarity for both general and domain-specific terms using synonyms from the WordNet hierarchical concepts (I. Park & Yoon, 2014). SAOs are also closely related to the functional concepts in the TRIZ framework, which uses the "problem-solution" models to mine TRIZ semantics(Abbas et al., 2014). These techniques have been used in fields such as the product forecasting, patent risk assessment, and competitive trend monitoring (Kim & Yoon, 2021; Yoon et al., 2013; Yoon & Kim, 2012a).

Despite these developments, SAO-based approaches still face a number of difficulties. Firstly, the accuracy of SAO element extraction across many sectors is still a major difficulty, despite the fact that the majority of SAO extraction methods rely significantly on intricate lexical and syntactic models. Additionally, the SAO processing is complicated by the significant semantic uncertainty (Sun et al., 2022). The inherent complexity of SAO triples limits its applicability in semantic similarity computations because there is no commonly accepted formula for determining SAO similarity, especially when the numerical variations of SAO triples in patents become conspicuously obvious (An et al., 2021; Chen et al., 2020; X. Wang et al., 2019). Furthermore, prepositions or prepositional phrases can be utilized to represent non-functional elements or relations that SAO triples usually ignore in the context of patents. For instance, the preposition "of" is used in the phrase "a feature of any object" to denote a subordinate relationship that no SAO structure can effectively capture. To overcome these limitations and improve context awareness, more knowledge-based extension tools have been created, including the WordNet dictionary (X. Li et al., 2020). Given the developments in the semantic analysis for practical patented applications, it is imperative to conduct a more thorough investigation into the technical aspects of patent solutions. Therefore, creating functional knowledge graphs that integrate many knowledge views is crucial to enhancing semantic analysis in patent systems.

### Patent Knowledge Graph

With the help of well-known information management and knowledge engineering projects like Google's knowledge vault, Wikidata, DBpedia, Freebase, YAGO, and others, knowledge graphs (KGs) have recently been established. The deep scientific knowledge mining depends on properly formatted citations and databases like the Microsoft Academic Graph, Scopus, Semantic Scholar, and OpenCitations, and therefore the efficient administration to facilitate the patent analysis makes a significant advance (Phyo, 2022). Patents require more attention from knowledge perspectives, especially with regards to invention levels, functional innovation, and economic potential, as over 90% of the most recent technical portfolios globally are patent knowledge, and 80% of these inventions were initially published (Li, Atherton and Harrison, 2014; An *et al.*, 2021). Thus, conventional knowledge graphs often fail to meet the special needs of patent semantic computation. Thus, the development of patent knowledge graphs has been the focus of a number of projects, often encompassing a variety of fundamental components, including information extraction, knowledge association, knowledge representation, data preprocessing, ontology design, and knowledge completion. Even though data preparation methods are fairly well-established, knowledge graphs still have space for significant advancements. IBM's India Research Lab created the BioPatent Miner system, which uses a biomedical semantic framework to provide related patent solutions (Niemann et al., 2017). The fine-grained patent knowledge extraction, such as the entity and relation extraction, is still ongoing. A promising technique for retrieving patent information has emerged that uses deep learning models for entity recognition and semantic relation extraction (Chen et al., 2020). Furthermore, a comprehensive examination of "Entity-Relation-Entity" sequences has been developed, integrating both functional and non-functional interactions together with an enhanced approach to calculating the similarity (An et al., 2021). Pre-trained and deep learning models may be useful to construct a patent semantic computing framework (Krestel et al., 2021). Additionally, some attempts have been made to create entity types and relation types in order to extract potential patent knowledge, with knowledge-based extension tools being positioned as necessary supplements (X. Li et al., 2020). Key product-related patent phrases were retrieved in order to increase the R&D effectiveness, where core components were then identified using a complicated network that used eigenvector centrality methods (Lin et al., 2022).

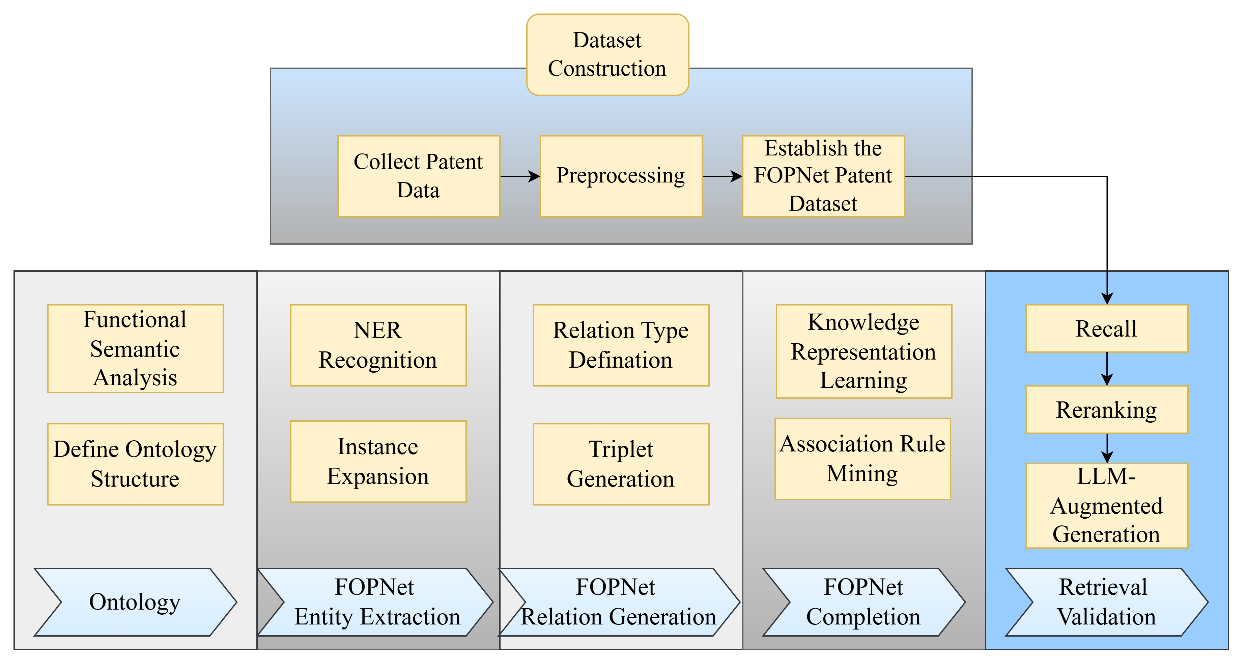
On the other hand, preliminary literature has also utilized TRIZ-based knowledge to guide the semantic analysis of patents (Cong & Tong, 2008). TRIZ itself has its own functional model’s definition, which has evolved and contributed to the development of USIT, TOP-TRIZ, and OSTM-TRIZ methodologies (Spreafico & Russo, 2016). Prior research has used ontology rules to extract domain-specific terms and create TRIZ-based product design systems in the combination of TRIZ and patent semantics. Object properties or attributes interacted to achieve a certain function, enriching the functional model with deeper meanings However, these systems' scalability and adaptability are limited by their reliance on handwritten rules (Chechurin, 2016; Spreafico and Russo, 2016; Ding *et al.*, 2017). Functional structures described by patents can be constructed by identifying technical features from the claims, and a personalized semantic TRIZ framework was built (Wodehouse et al., 2017). Through the pattern analysis, semantic TRIZ elements extracted from patents, led to the creation of a functional structure database with more than 400 functional categories serving as a guide for the patent analysis (Yan et al., 2014; Ding et al., 2017). Other functional modeling widely recognized as the Functional Decomposition and Morphology (FDM), helps to highlight the functional relationships between system components (Fiorineschi et al., 2020). These achievements in the functional analysis have led to the development of entity and relation characteristics within patents (Fiorineschi et al., 2020). Approaches related to the modern TRIZ are increasingly focusing on the functional features of patent texts, contributing to a deeper understanding of functional-centric semantic characteristics (Guarino et al., 2022).

These aforementioned attempts of abstracting functional concepts that align with TRIZ expertise or functional models, the approaches have produced little achievements. Relatively, few systematic debates have yet been addressed on the functional semantic features of patents. More effective techniques for semantic feature integration should also be thoroughly examined in order to strengthen the basis for the patent analysis (Phyo, 2022). Therefore, we have developed and constructed a practical system, FOPNet, that combines TRIZ principles with patent knowledge graphs while accounting for the nuances and characteristics of patents in order to enhance the semantic understanding.

## Methodology

This FOPNet methodology consists of the ontology design, entity extraction, relation generation, knowledge completion and validation modules. As shown in Figure 1 below, a complete illustration of the methodology has been provided. Firstly, the concepts, entities, and relations related to the patent contents are specified by a well-defined ontology named the functional semantic ontology which is derived from the TRIZ functional modeling. Key entities, comprising the Component, Function (F), Object (O), and Property (P), are defined and substantively identified from the patents using the named entity recognazition (NER) algorithms in the FOPNet Entity Extraction process. The next instance extension process aligned with the current ontological structure by adding or expanding entity synonyms and hypernyms to provide a thorough semantic coverage of entities. Furthermore, relation types of recognized entities are specified and categorized using pre-trained models (PTMs) and deep learning algorithms, setting the stage for the later knowledge graph construction. Then patent documents are parsed into organized semantic entities and relations, such as numerous ⟨F, O, P⟩ triplets (i.e. FOPNet triplets) or other multi-tuple forms. Moreover, missing or ambiguous elements of FOPNet triplets should be identified and complemented using the required knowledge completion process.

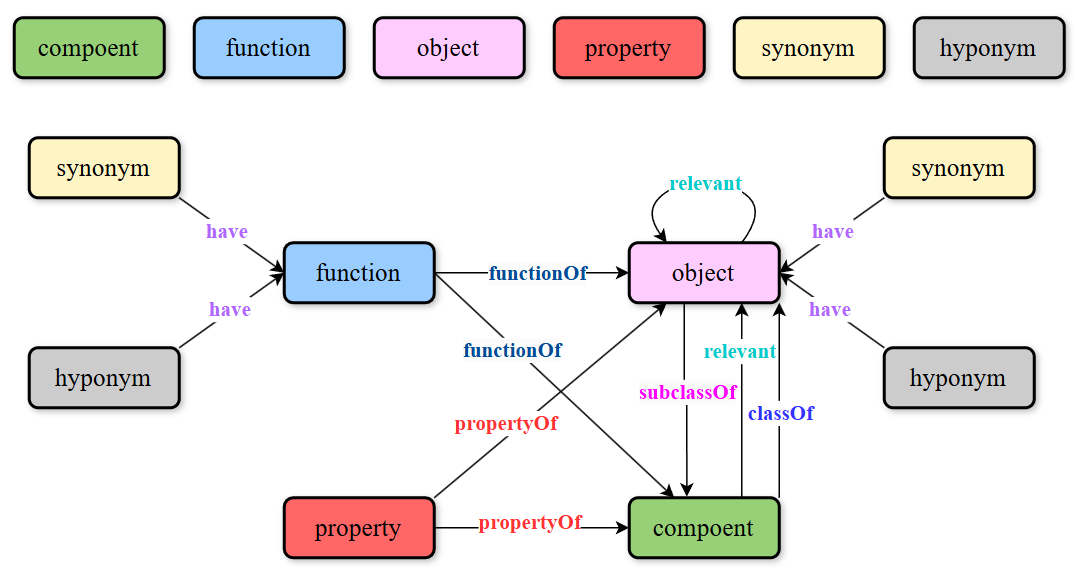
To improve the FOPNet completeness, key groupings or possible relationships should be found. Therefore, the FOPNet is then completed by the knowledge representation learning algorithms which transform entities and relationships into continuous vectors so the models can infer associations that are not explicitly represented. The FOPNet is thus able to build multi-dimensional links by automatically identifying entities that are functionally closely related or objects that are commonly associated through the use of association rule mining or community detection methods. Lastly, a comprehensive retrieval application is constructed to validate the FOPNet effectiveness and efficiency in real-world scenarios. During the validation phase, the system replicates the semantic search in real-world contexts using knowledge matching metrics, recall and reranking algorithms, and large language model (LLM) augmented techniques.



**Figure 1. Flowchart of the FOPNet methodology.**

### Functional Semantic Ontology

A new patent ontology called the functional semantic ontology has been developed that illustrates the features of patent technology from both a macro and micro viewpoint. The novel ontology design distinguishes between hierarchical and synonymy links among words (Zhai et al., 2022). The ontology serves as a framework for organizing information and building the functional semantic structure, which describes the semantic relations of patents, by creating concepts, a collection of node types, and elements that represent entities like objects, functions, properties, and components, as seen in Figure 2. Through such a unified semantic network, the ontology abstracts the "functional logic" presented in patents into basic entity and relation types, simplifying the semantic structure and enhancing the clarity of relationships. These colored essential features are often dispersed across claim sections, forming complex semantic hierarchies and a range of interactions labeled on the connecting lines (i.e. functionOf, propertyOf, relevant, have, classOf, subclassOf). By mapping each technological solution described in a patent to the relevant entities together with their auxiliary concepts (Relation, Synonym, Hyponym) at the ontology level, a coherent structural foundation for the subsequent technical analysis would be addressed.



**Figure 2 Structural Diagram of the FOPNet Ontology Construction.**

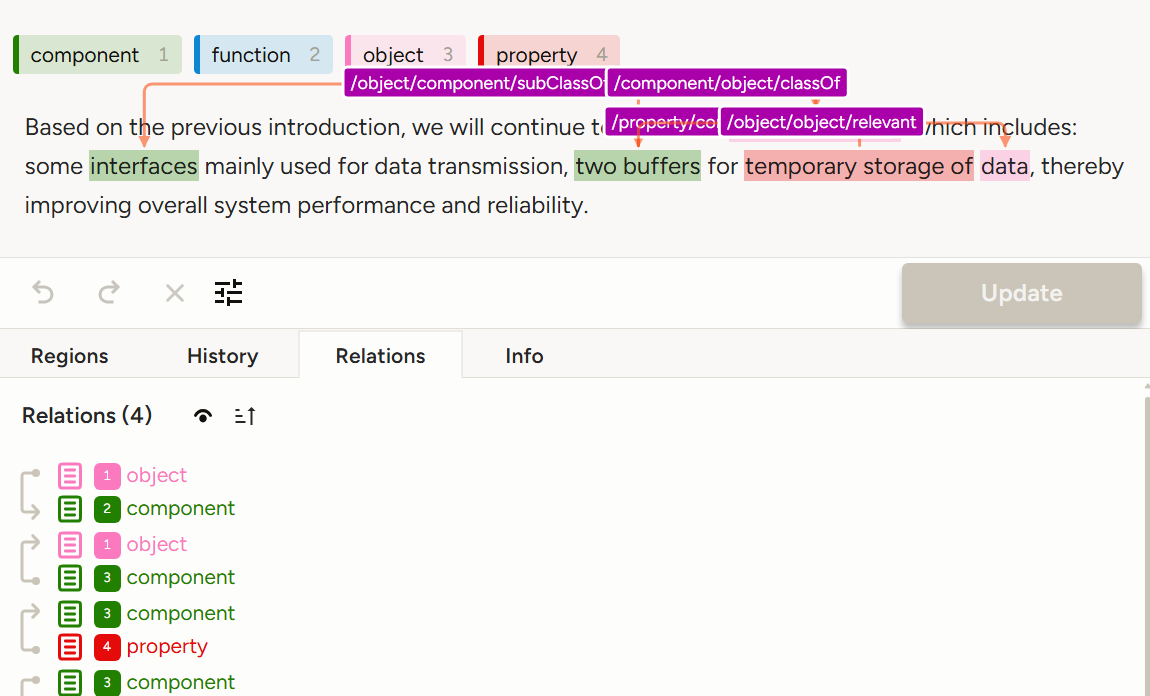
Notes: Component (Green) are specific parts or subsystems that are involved in technical solutions. Function (Blue) represents the specific function; Object (Pink) is that the entity or target upon which the function is implemented. Property (Red) describes the object or function characteristics. Synonym (Yellow) represents semantic extensions of functions or objects; Hyponym (Gray) is the hierarchical refinement of functions or objects.

### Data Processing Flow

Amounts of data have been made publicly available, and the overall number of patent applications filed with the USPTO has expanded dramatically during the last few decades[[1]](#footnote-1). To meet the successive requirements, a sizable and diverse collection of semi-structured patent data has been compiled and preprocessed to extract key contents, such as abstracts, titles, and keywords, which are then covered in the parsed patent documents. Then, the Label Studio Toolkit[[2]](#footnote-2) was utilized to automatically annotate the data. Frequent semantic types identified in patent literature by parsing the patent claims have been constructed using the functional semantic ontology shown in Figure 2. The remote supervision assumption was carried out that the other unannotated sentences are initially constructed around small FOPNet triplets, especially the F, O, and P groups as the seeds. In order to maximize and expand the original annotations, domain-specific trigger words (such "for," "having," "belonging to," etc.) were also established, and rules based on syntactic dependency relations were created:

|  |
| --- |
| {  "na": 0,  "/component/object/classOf": 1,  "/object/component/subClassOf": 2,  "/function/component/functionOf": 3,  "/function/object/functionOf": 4,  "/property/component/propertyOf": 5,  "/property/object/propertyOf": 6,  "/component/object/relevant": 7,  "/object/object/relevant": 8  } |

During the post-processing stage, annotated entities and the relationships between datasets are included in the results. An expected output format is shown in Figure 3. The annotated entities and related relations are included. Rule matching and greedy matching approaches would be used to improve and expand on the first annotation results. Rule matching creates relationships between entity pairs by examining linguistic patterns. For instance, 'for' normally suggests a 'functionOf' relationship between an object and a function, whereas 'have' usually denotes a 'ownerOf' relationship between an object and a property. Preliminary semantic linkages between entity pairs are established by combining it with greedy matching technology to find the closest relationship trigger words. The ability to swiftly and precisely identify these trigger words—such as "functionOf," "propertyOf," etc.—is essential for creating a knowledge graph since they specify the kind of link between entities. The final annotated data was saved in JSON format for the future processing and model training if the manual verification of the annotated results completed.



**Figure 3. GUI display of the FOPNet annotation using the Label Studio Tool**

### Entity Recognition and Extension

Named Entity Recognition (NER) is a foundational step in the development of FOPNet that assists in identifying significant items in sentences, which provides crucial contextual information of patents. Traditional NER algorithms can be divided into three categories: rule-based, statistical, and deep learning methods. Rule-based approaches relied on manually constructed rules and pattern matching, which worked well for structured data. However, as time went on, statistical machine learning techniques were increasingly applied to NER problems (Ma & Hovy, 2016). Deep learning-based NER approaches have gained more importance, especially those that use pre-trained language models like Bidirectional Encoder Representations from Transformers (BERT) and Bi-directional LSTM and Conditional Random Fields (BiLSTM-CRF) (Devlin et al., 2019). An entity recognition model has been used to identify significant entities in the patent literature to ensure the logical consistency and scientific validity of the FOPNet (Teng et al., 2024). The functional entity (Function) typically appears as a verb or verb phrase in patents, while the object entity (Object) is typically considered as the object of the verb's direct action with the property (Property) description. Methods such as term sorting, type correction and deduplication were further employed to refine the results and increase the NER accuracy of the FOPNet.

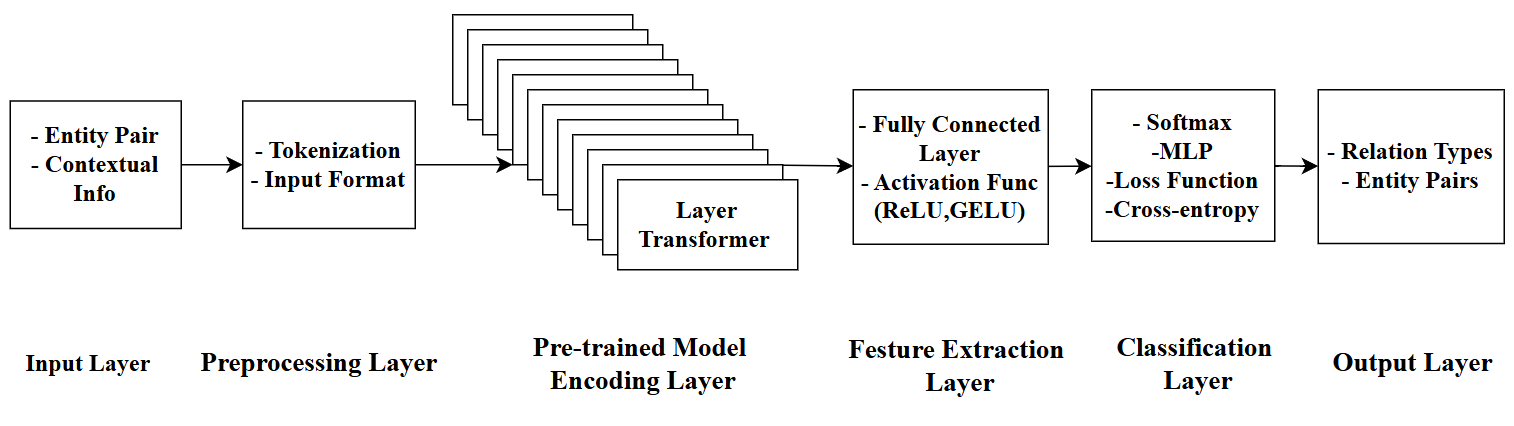
The patents' semantic coverage and domain relevance can be further enhanced by applying the WordNet extension technique. Using lexical semantic relationships, FOPNet terms' synonyms and hyponyms could be added. Words are sorted and extended into synsets using the WordNet library, and each synset signifies a collection of concepts that are synonymous or closely related. Synonyms of FOPNet entities could be increased by searching the WordNet, which would boost the accuracy of the subsequent entity matching. In order to help locate concepts into more specific categories, the WordNet also offers hierarchical links. As an illustration, "dog" is a hyponym of "animal," and yet "poodle" is similar to "dog." The subordinate terms of FOPNet entities can be retrieved using the process below, expanding the knowledge hierarchy and improving the capacity to understand finer-grained concepts and perform the semantic analysis. The following algorithms have outlined the procedures for expanding synonyms and hyponyms for FOPNet entities:

|  |
| --- |
| **Algorithm 1: Synonym Expansion**  Input: A list of fop\_entities.  Output: A dictionary expanded\_entities where each key is an entity and its value is a set of synonyms.  *expanded\_entities ← empty dictionary*  *for each entity in fop\_entities:*  *synsets ← get all synsets for entity*  *synonyms ← empty set*  *for each synset in synsets:*  *for each lemma in synset.lemmas():*  *synonyms ← synonyms ∪ {lemma.name()}*  *expanded\_entities[entity] ← synonyms*  *return expanded\_entities*  **Algorithm 2: Hyponym Expansion**  Input: A list of fop\_entities.  Output: A dictionary expanded\_entities where each key is an entity and its value is a set of hyponyms.  *expanded\_entities ← empty dictionary*  *for each entity in fop\_entities:*  *synsets ← get all synsets for entity*  *hyponyms ← empty set*  *for each synset in synsets:*  *hyponyms\_of\_synset ← get all hyponyms of synset*  *hyponyms ← hyponyms ∪ hyponyms\_of\_synset*  *expanded\_entities[entity] ← hyponyms*  *return expanded\_entities* |

### FOPNet Relation Extraction

The relation extraction task is to identify semantic ties between items in patents. Most of the datasets consist of entity pairs with their relations, often containing contextual information about the entities in a sentence fragment. Recently, the most widely used methods often use models based on BERT or GPT for the relation categorization or relation prediction (Beltagy et al., 2019; Devlin et al., 2019). Deep semantic representations were customized by these models with the use of pre-trained tasks, providing a strong base for fine-tuning tasks. An overview of the FOPNet relation extraction workflow is shown in Figure 4. Pre-trained models (PTM) can be dynamically replaced to provide strong semantic capabilities. These models can then be modified to meet specific categorization requirements, enhancing their generalization ability and overall performance(Raffel et al., 2020). Output features from the pre-trained models are transferred to a higher-dimensional space by the fully connected layer, which qualifies them for the ensuing classification tasks. Based on the features that have been obtained, the entity relationship is predicted using the classification layer. With corresponding entity pairings, the outputs are often classified according to relation kinds. Then, the FOPNet can be implemented using most forms of entity relations.

A range of pre-trained models can be employed to achieve this goal of the relation extraction, which captures contextual information to perform remarkably well in relation extraction tasks despite using a lot of computing resources. SciBERT is more effective in extracting scientific text terminology and relationships because it is a domain-specific variant of BERT that has already been trained on scientific literature. In constrast, TinyBERT is a lightweight distillation-based version of BERT that uses less processing resources while still performing well in relation extraction tasks. In 2018, OpenAI introduced the Transformer-based autoregressive language model named GPT, which analyzed texts from left to right, and generated coherent and flowing sequences. Additionally, the T5 (Text-to-Text Transfer Transformer) sequence-to-sequence model was also utilized, which flexibly treated the relation extraction as text transformations without additional requirements. The PTM-based relation extraction methods, which concentrate on categorized relationships at the technical function level, come after this procedure. For example, FO (Function-Object), OF (Object-Function), FOP (Function-Object-Property), OFP (Object-Function-Property), and FP (Function-Property) are the ordinary kinds of relations that make up the "relevant" type.



**Figure 4. Flowchart of FOPNet Relation Extraction**

### FOPNet Knowledge Completion

Uncertain semantics or missing links may arise from patent circumstances containing incomplete definitions of functions (Function), objects (Object), or properties (Property). As a result, inadequate knowledge becomes a common issue while developing the FOPNet. One major challenge is determining how to automatically infer probable functional semantic correlations from existing data. Various reasoning strategies, known as rule-based reasoning, path-based reasoning, and knowledge representation learning, have been employed to develop current knowledge completion systems. Additionally, some approaches and materials have emerged to enhance knowledge graphs' completeness and resilience. Herein, knowledge representation learning and association rule mining approaches are used to fulfill the FOPNet.

Knowledge Graph Embedding (KGE) techniques, which represent entities and relationships as low-dimensional vectors, are now the primary methods used in knowledge representation learning to extract pertinent high-dimensional sparse information. Through the embedding vectors of objects and relationships into continuous vector spaces, the representation learning facilitates the knowledge reasoning and completion. Several well-known benchmarks, including the AIDA, MAG, FB15K and WN18, have garnered significant interest in translation-based learning models, such as those offered by TransE, TransH, RotatE, QuatE, HAKE and ComplEx[[3]](#footnote-3). With **ℎ** representing the head entity, **𝑟** the relationship, and 𝑡 the tail entity, these models guarantee that the triplet (***h, r, t***) fits the equation **ℎ+𝑟≈𝑡**, and make it feasible to portray the semantic linkages between entities more effectively. Among these models, the TransE model is one of the earliest and most fundamental models. Assuming that for a valid triplet (***h,r,t***), the embedding of the tail entity ***t*** should be close to ***h+r***. This struggles with more complex relational patterns even if simple and efficient for one-to-one relations. In contrast, the TransH model improved its capacity to describe many-to-many or hierarchical relations by allowing entities to have distinct representations for various relations by projecting them onto relation-specific hyperplanes. The ComplEx model also leveraged complex-valued embeddings and particularly exceled at modeling asymmetric relations, making it suitable for large-scale knowledge graphs. The RotatE introduced a more expressive mechanism by treating relations as rotations in the complex vector space, capturing various relation patterns like symmetry, antisymmetry and inversion. Extending this idea, the QuatE used quaternions (four-dimensional numbers) for embedding entities and relations, enabling it to represent even more intricate semantic interactions across dimensions.

Moreover, the multi-relation reasoning processes can be used to further enhance their ability to construct large knowledge networks. In the FOPNet, the relationships between Function, Object, and Property entites can be effectively modeled, which further enriches entity representations by aggregating information from neighboring entities, could improve the knowledge completion. Missing or implicit connections among entities and relations can be predicted by employing such an trained model. The following process demonstrates how representation learning models can be applied for the FOPNet completion:

|  |
| --- |
| Input: Knowledge graph KG = (E, R, T) where E = entities, R = relations, T = triplets,Number of training epochs,Learning rate α  Output: Learned entity embeddings ***E\_emb***,Learned relation embeddings ***R\_emb***  **Step 1: Embedding Initialization**  *function INITIALIZE\_EMBEDDINGS(KG):*  ***E\_emb*** *← empty dictionary*  ***R\_emb*** *← empty dictionary*  *for each entity e in KG.E do*  ***E\_emb[e]*** *←* ***random******vector*** *∈ ℝ^d*  *for each relation r in KG.R do*  ***R\_emb[r]*** *←* ***random******vector*** *∈ ℝ^d*  *return* ***E\_emb****,* ***R\_emb***  **Step 2: Loss Computation**  *function CALCULATE\_LOSS(h, r, t,* ***E\_emb****,* ***R\_emb****):*  *return ‖****E\_emb[h]*** *+* ***R\_emb[r]*** *-* ***E\_emb[t]****‖₂*  **Step 3: Training Process**  *function TRAIN(KG, epochs, α):*  ***E\_emb****,* ***R\_emb*** *← INITIALIZE\_EMBEDDINGS(KG)*  *for epoch = 1 to epochs do*  *total\_loss ← 0*  *for each triplet (h, r, t) in KG.T do*  *loss ← CALCULATE\_LOSS(h, r, t,* ***E\_emb****,* ***R\_emb****)*  *total\_loss ← total\_loss + loss*  *# Gradient descent updates*  ***E\_emb[h]*** *←* ***E\_emb[h]*** *- α∇****E\_****embh*  ***R\_emb[r]*** *←* ***R\_emb[r]*** *- α∇****R\_embr***  ***E\_emb[t]*** *←* ***E\_emb[t]*** *- α∇****E\_embt***  *print("Epoch", epoch, "Loss:", total\_loss)*  *return* ***E\_emb****,* ***R\_emb***  **Step 4: Relation Prediction**  *function INFER\_RELATION(h, t,* ***E\_emb****,* ***R\_emb****):*  *return* argmin\_r *‖****E\_emb[h]*** *+* ***R\_emb[r]*** *-* ***E\_emb[t]****‖₂* |

Moreover, association rule mining techniques are also essential to the FOPNet knowledge mining in order to uncover potential relationships between knowledge nodes from a large volume of transactional data. As the amount of data increases, the classical Apriori technique becomes unsuitable for real-world applications due to its shortcomings, such as the need for many database scans and the generation of a large number of candidate itemsets. To fully construct FOPNet, two conventional association rule mining algorithms, FP-Growth[[4]](#footnote-4) and FP-MAX[[5]](#footnote-5), were selected for the efficiency of managing enormous datasets. The FP-Growth method compresses raw data using a "divide and conquer" strategy by building a small FP tree structure, which only requires two database scanning methods: one to build the FP tree and another to check whether each item is supported. To be further, conditional FP trees and conditional pattern bases are created recursively to mine every frequently occurring itemset. The main advantage of the FP-Growth is that, by avoiding the development of candidate itemsets, it drastically reduces the computational cost. In constrast, the FP-MAX algorithm has put more concentration on the most frequent itemset discovery, including another pruning technique based on the FP tree, which increases the efficiency by detecting and trimming search routes for early non-maximum itemsets. The FP-MAX is particularly well-suited for processing large datasets with complex patterns and can significantly reduce the memory utilization. Implementing the effectiveness of two association mining approaches, the FOPNet knowledge correction and completion will be incorporated as seen below:

|  |
| --- |
| **## FP-Growth Algorithm** Input: Transaction database D, minimum support threshold min\_Sup Output: All frequent itemsets *Calculate the support for each item Remove items that have support less than min\_sup Sort the remaining items in descending order of support Construct the FP-tree:  Create the root node of the FP-tree, labeled as null  For each transaction in the database:  - Remove infrequent items  - Sort the remaining items in descending order  - Insert the sorted items into the FP-tree Extract frequent itemsets from the FP-tree:  For each frequent item:  - Construct the conditional pattern base  - Construct the conditional FP-tree  - Recursively extract frequent itemsets from the conditional FP-tree Return all frequent itemsets* **## FP-MAX Algorithm**  Input: Transaction database D, minimum support threshold min\_sup Output: All maximal frequent itemsets *Construct the FP-tree:  Create the root node of the FP-tree, labeled as null  For each transaction in the database:  - Remove infrequent items  - Sort the remaining items in descending order of support  - Insert the sorted items into the FP-tree Extract maximal frequent itemsets from the FP-tree:  For each frequent item:  - Construct the conditional pattern base  - Construct the conditional FP-tree  - Recursively extract maximal frequent itemsets from the conditional FP-tree Return all maximal frequent itemsets* |

## Results and Discussions

### Entity Recognition Results

The aim of these studies is to thoroughly assess the suggested models' entity recognition capabilities. Standard measures, such as F1-score, Precision, and Recall, were used to evaluate the performance. To give a more comprehensive examination of the outcomes, the weighted average of these measures was also computed. The recognition performance for each type of entity is shown separately in Table 1. The weighted average scores of 0.85, 0.89, and 0.87 show that all entity types perform similarly, according to the data.

Semantic extension improves the links between hyponym and synonyms, giving significant support for semantic matching at the conceptual and instance levels. This also encourages more thorough linkages.The hyponyms, synonyms, and hypernyms of the identified items were extracted when the WordNet program was applied.As shown in Table 2, the introduction of entity extensions considerably helps the interpretation and analysis of technical literature. This approach helps uncover broader variants and finer semantic hierarchies in patent material. The data in Table 2 displays the effects of expanding functional and attribute entities with FOPNet, including synonym extension and subword extension. These extension methods help strengthen the semantic expression ability of entities and further improve the information retrieval and analysis performance in technical literature. For example, subjunctive extensions can identify more particular actions or scenes, such as "constitute" or "position," whereas functional verbs like "compute" can be coupled with broader notions, such as "include" and "be."

**Table 1 Different entity recognition results using the proposed NER method**

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity Types** | **Precision** | **Recall** | **F1-score** |
| Component | 0.82 | 0.81 | 0.82 |
| Object | 0.87 | 0.87 | 0.87 |
| Function | 0.81 | 0.90 | 0.85 |
| Property | 0.87 | 0.92 | 0.90 |
| Weighted Avg | 0.85 | 0.89 | 0.87 |

**Table 2 Entity extention results of the FOPNet.**

|  |  |
| --- | --- |
| **Entity Manipulation** | **Entities and the extentions** |
| **Function: comprising** | |
| **Synonym Expansion** | {'make\_up', 'constitute', 'be', 'contain', 'incorporate', 'represent', 'consist', 'comprise'} |
| **Hyponym Expansion** | {'constitute', 'supplement', 'fall\_into', 'present', 'pose', 'fall\_under', 'straddle', 'compose', 'form', 'make', 'range'} |
| **Property: resolution** | |
| **Synonym Expansion** | {'solvent', 'firmness\_of\_purpose', 'resolution', 'firmness', 'resoluteness', 'result', 'answer', 'declaration', 'solution', 'resolving\_power', 'closure', 'resolve', 'solving', 'resolving', 'settlement'} |
| **Hyponym Expansion** | {'possession', 'denouement', 'bullheadedness', 'stubbornness', 'willpower', 'obstinacy', 'self-will', 'stiffness', 'diagonalization', 'factoring', 'adamance', 'factorization', 'sturdiness', 'factorisation', 'determination', 'purpose', 'decision', 'joint\_resolution', 'decisiveness', 'steadiness', 'diagonalisation', 'self-possession', 'obduracy', 'will\_power', 'steadfastness', 'single-mindedness', 'pigheadedness', 'self-command', 'self-control', 'obstinance', 'unyieldingness'} |

Concept level matching, as seen below, is based on semantic hierarchy and aims to expand lower and higher level terms in order to provide a wider range of relationships. Instance level matching will focus on specific semantic instances, leveraging synonym extensions to improve matching accuracy and help algorithms in uncovering relevant meanings. Furthermore, instance level matching has eased cross-domain matching based on semantic similarity while generating more accurate results for practical applications. The various FOPNet knowledge matching levels are as follows:

|  |
| --- |
| **Example: Concept-Level Matching**  Assuming we encounter two root words, "calculating" and "resolution", while conducting patent information retrieval, the system can identify them through concept level matching.  *"Computing" is associated with its upper words like "or" and "be", so it can be more broadly matched to synonyms such as "contain" and "representative".*  *The upper level words of "resolution" include more abstract concepts such as "decision", "cognitive process", etc., which can be matched across different domains.*  **Example: Instance-Level Matching**  For example, in the case of "computing":  *The homophones can be extended to "make up", "constitution", "be", etc., which can help us connect "computing" with these synonymous phrases in patent literature, especially in technical descriptions, and match them to specific technical steps or methods.*  *For "resolution", synonym extensions include "solution", "declaration", "settlement", etc. These synonymous phrases can be used to accurately identify solutions or decision-making processes in technical literature, improving the accuracy of semantic understanding.* |

### Relation Extraction Results

To develop the FOPNet, the semantic links between things must be adequately defined. Due to a considerable imbalance in sample quantities between relation types, the Micro F1 and AUC were selected as evaluation measures to give a comprehensive assessment of the performance of various PTM-based models in multi-class classification tasks.

Table 3 provides a concise review of the experimental results. The Micro F1 evaluates the overall effectiveness of the multi-class classification, whereas the AUC measures a model's ability to distinguish between the presence and absence of relationships. While an increase of Micro F1 indicates a more consistent overall performance in the multi-category connection extraction, a higher AUC value depicts a strong ability of models to differentiate between positive and negative categories.

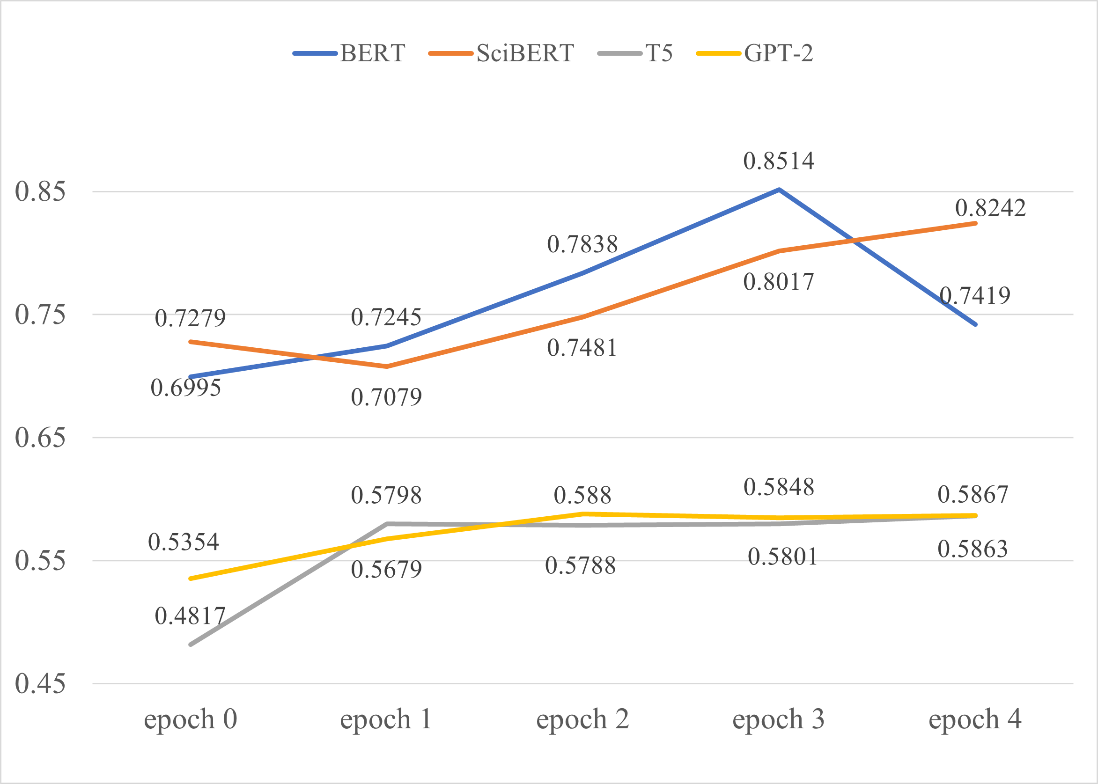
**Table 3 Relation extraction results using different pre-trained models.**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model Configurations** | | | | **Before Data Augmentation (100 Entries)** | | | | **After Data Augmentation (200 Entries)** | | | |
| Pre-trained Models | batch\_size | | max\_epoch | AUC | | Micro F1 | | AUC | | Micro F1 | |
| Scibert | 16  16  16  16 | 5 | | | 0.4521 | | 0.6749 | | **0.733** | **0.7419** |
| Bert | 5 | | | 0.4764 | | 0.678 | | 0.7079 | 0.7403 |
| T5 | 5 | | | 0.3912 | | 0.6723 | | 0.6103 | 0.6952 |
| GPT-2 | 5 | | | 0.3886 | | 0.6667 | | 0.5863 | 0.6809 |

The efficacy of numerous PTM-based models throughout several training cycles has been illustrated in Figure 5. When training data quantities rose, the AUC and Micro F1 metrics of all models could be significantly improved. This improvement demonstrates that the model can learn additional contextual knowledge, improving its ability to classify relations. The preliminary results have shown that best-performing models among those could be BERT-based and SciBERT-based models, which have also demonstrated particularly noteworthy increases with more training data. Constratively, the T5-based and GPT-2-based models performed slightly worse although they performed better with more data, and of course, the differences among these models were significantly smaller. Although additional data increased their performances, it was significantly less than those of the BERT and SciBERT models. Overall, the AUC and Micro F1 metrics of each model have increased significantly with the amount of training data.

While BERT attained the greatest peak Micro-F1 score of 0.8514 in the third epoch, its performance declined dramatically in the fourth epoch due to potential learning rate instability. In contrast, SciBERT demonstrated continuous and constant improvements, reaching 0.8242 in the fourth epoch, virtually matching BERT’s peak. The T5 model experienced variations but earned its best score of 0.8514 in the third epoch, demonstrating potential for certain tasks. GPT-2, however, followed behind, with minimal progress and a poorer overall performance, making it a secondary alternative. These findings illustrate SciBERT’s robustness and BERT’s potential for peak performance given adequate training settings.

This advancement indicates that Bert-based models can obtain more contextual knowledge to improve their relation classification abilities, implying that they are more effective and particularly well-suited to dealing with difficult semantic modeling difficulties in relation extraction tasks.

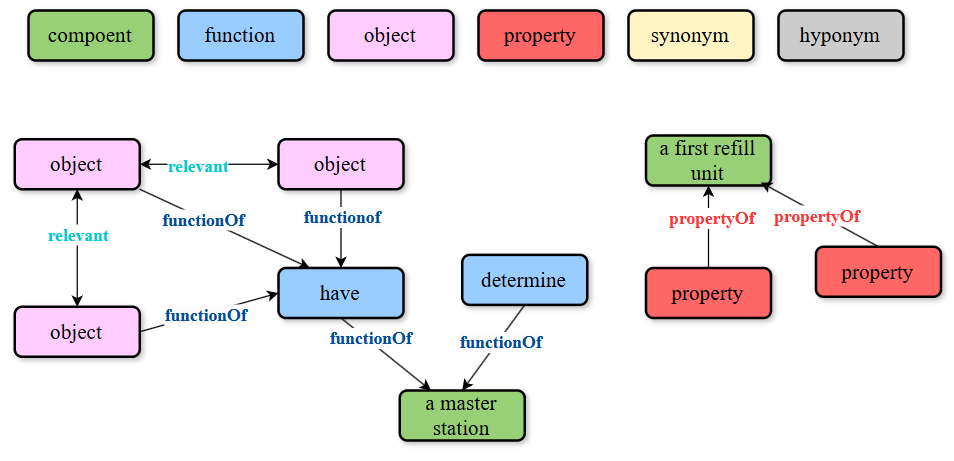


**Figure 5. Performance results of the multiple training epochs of these models.**

Entity nodes, related edges, and relevant extended nodes can all help to complete the process of developing a patent knowledge graph. The graph development method consists of two basic steps: importing entities (represented by blue,red, purple, and green nodes) and establishing relation edges to connect the respective nodes. By merging these nodes and edges in this preliminary result, a clear and understandable depiction of the overall structure of a patented claim could be obtained. To illustrate this logic, an exemplified patent claim was imported below:

"*A system for managing soap, sanitizer, and lotion refills comprising: a first dispenser; the first dispenser having a first dispenser processor; a first dispenser memory; first dispenser communications circuitry; a first refill unit associated with the first refill dispenser; a master station; the master station having a master station processor; a master station memory; master station communications circuitry; logic stored in one of the first dispenser memory and the master station memory for determining one or more usage rates of the first refill unit in the first dispenser; logic stored in one of the first dispenser memory and the master station memory determining an expected depletion date for the first refill unit as a function of one or more usage rates for the first dispenser*."

This result has been analyzed using the aformentioned relation extraction approaches to identify significant entities, relations, or other features in example. These items into a knowledge graph, with nodes representing entities and edges denoting their relationships. .The result of the relation extraction process, as shown in Figure 6, has been organized into a knowledge graph where nodes represent different entities (e.g., "object," "component," "function," "property") and edges denote the relationships between them.



**Figure 6. Entity & Relation Knowledge Extraction of a patent.**

### Knowledge Completion Results

To achieve the constraints of knowledge graph completeness, knowledge representation learning techniques are first used to forecast potential entities or relationships. Structured knowledge can be efficiently encoded into low-dimensional vector spaces using a variety of models and techniques to manage entity and relationship embeddings in knowledge graphs (KGs). Some open-source toolboxes for knowledge representation learning (KRL), such as NeuralKG[[6]](#footnote-6), make it easier to complete subsequent tasks as the link prediction and knowledge graph completion. The FOPNet KRL datasets were manipulated and used for the training and assessment, which contained a lot of entities and relations suitable for assessing how well KRL models work and would also be publicly accessible. The following are the precise experimental steps:

1. Defining the loss function, wherein the embedding is optimized using a marginal based loss function with *L2* normalization to guarantee that effective triples receive higher scores than damaged triples;
2. Embedding initialization, wherein entity and relationship embeddings are initialized as randomly generated vectors in a continuous vector space;
3. Training the models using gradient descent optimization, which iteratively updates the embedding to minimize the loss function;
4. Assessing the performance of the trained models on the validation dataset using link prediction metrics, such as mean reciprocal ranking (MRR), and hit rates for topN ranking (Hits@N).

The outcomes of merging the NeuralKG project with the first, second, and third datasets are compiled in Table 4. These datasets relate to various knowledge graph features: The primary goal of First\_data is to extract items like "functions," "objects," or "attributes," Relational words, such as "subnyms" and "synonyms," are represented by Second\_data; an entity in First\_data that has a comparable relationship with an entity in Second\_data is called Third\_data. According to the experimental findings, Third\_data performed optimally across all models, with ComplEx achieving the greatest MRR of 0.566. This suggests that Third\_data has a wide range of relational patterns, such as symmetric or antisymmetric interactions that favor Rotate, bidirectional semantics appropriate for ComplEx, and hierarchical structures in line with HAKE. Stronger learning signals are produced by Third\_data's richness, which enables the model to take advantage of its complexity to perform better. ComplEx excels in Hit@N in particular. It has been shown to be capable of handling datasets of great complexity, ranging from 13.8% of First\_data to 74.9% of Third\_data.On the other hand, Second\_data performs relatively well due to its somewhat complex relationships, like hierarchical linking or partial symmetry. Even though HAKE's MRR of 0.357 on Second\_data was marginally higher than ComplEx's 0.345, the relational schema's lower complexity makes the overall performance less important than Third\_data. The model's overall performance is lowest on First\_data, which represents isolated entities, since the dataset does not have the relational complexity needed to supply enough learning signals. With an MRR of 0.186, HAKE excels on First\_data due to its hierarchical nature, which enables it to efficiently capture basic patterns. However, the performance of all models is naturally constrained by the dataset's simplicity. ComplEx is able to handle complex relationships, which allows it to perform the best among these models on complex datasets like Third\_data. In the same way, RotatE is very flexible and obtained an MRR of 0.530 on Third\_data, which is similar to ComplEx in the following ways. Hit@N Although its hierarchical structure has been shown to be beneficial, the score HAKE stands out on simpler datasets like First\_data. However, as the dataset's complexity rises, its performance steadily gets better. Contrarily, TransE has continuously underperformed across all datasets, generating usually low results because to its inability to represent intricate interactions like many-to-many linkages.

These findings highlight how crucial dataset complexity is for knowledge graph embedding tasks. Even while Third\_data offers the most context and produces the greatest overall outcomes, performance optimization greatly depends on the model selection. HAKE[[7]](#footnote-7) does excellent in simpler circumstances, whereas ComplEx and Rotate show significant capability on complex datasets. TransE, however, ran into issues on every dataset, underscoring the necessity for more sophisticated techniques to manage intricate interactions.

Table 5, which shows partial completion results for FOP triplets utilizing representation learning algorithms, illustrates how well the model handles missing triplet components, such as the head, tail, or relationship.

**Table 4 Prediction results of different models at triplet missing positions**

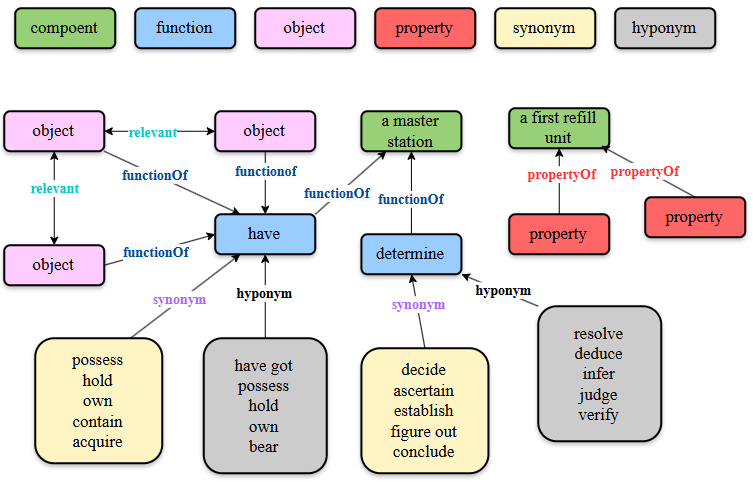
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Metrics** | **First\_data** | **Second\_data** | **Third\_data** |
| TransE | | Hit@10 | 0.13799 | 0.14699 | 0.14699 |
| Hit50 | 0.13899 | 0.14800 | 0.14800 |
| Hit100 | 0.13899 | 0.14800 | 0.14800 |
| MRR | 0.13799 | 0.14699 | 0.14699 |
| ComplEx | Hit10 | 0.13799 | 0.48300 | 0.74900 |
| Hit50 | 0.13899 | 0.60500 | 0.84100 |
| Hit100 | 0.13899 | 0.65700 | 0.8700 |
| Mrr | 0.13799 | 0.34499 | 0.56599 |
| RotatE | Hit10 | 0.14000 | 0.43500 | 0.71200 |
| Hit50 | 0.14800 | 0.54199 | 0.81199 |
| Hit100 | 0.15800 | 0.59200 | 0.84399 |
| Mrr | 0.13899 | 0.32699 | 0.52999 |
| HAKE | | | Hit10 | 0.22499 | 0.46700 | 0.60600 |
| Hit50 | 0.28400 | 0.56199 | 0.72299 |
| Hit100 | 0.32100 | 0.60799 | 0.68599 |
| Mrr | 0.18600 | 0.35699 | 0.46500 |

**Table 5 Knowledge representation learing results for the FOPNet completion types**

|  |  |
| --- | --- |
| **Types** | Input: [street\_corner,synonym, 0] |
| **Tail completion** | The first 5 predicted tail entities are: [street\_corner,courtesy, git,hat,aromatic] |
| **Relationship completion** | Input: [horizontal\_surface,0,paved\_surface] |
| The predicted relationship is: [hypernym] |
| **Head Completion** | Input:[0,hyponym,traverse] |
| The first 5 predicted head entities are:  [food\_stamp, gauzy,tumult,narrow\_escape,vegetable] |

Furthermore, the same preprocessed dataset was used for the independent execution of the FP-MAX and FP-Growth algorithms with the same hardware setups and parameter settings. A dataset of several transactions was used in the experiment; each transaction included a set of objects, such as a product and one or more components. In order to maintain data correctness and completeness, the dataset was preprocessed to assure data quality. This involved removing irrelevant and noisy data as well as incomplete or irrelevant information. To make algorithm processing easier, the data was also transformed into a standardized format. For comparative analysis, a controlled experimental technique was employed.

According to the experimental findings, these two techniques' performance hardly changes at all. The processing time of FP-MAX is marginally quicker than that of FP Growth at a lower size (1000 records). This performance advantage, however, was greatly diminished when FP-MAX shown a decline in efficiency when assessed using a bigger dataset (100000 records). For association mining jobs, the FP Growth algorithm is utilized. To finish our understanding, we shall keep using the sample sentences from the earlier text. Enhancing the returned FOPNet can be achieved by completing the knowledge created in the previous text, such as by giving "have" and "determine" synonyms and subnyms. This fact demonstrates that the chosen procedure is realistic, and the processing output is unaltered because there aren't any frequent itemsets in this example. Figure 7 displays FOPNet's final output following the association mining and representation learning processes.

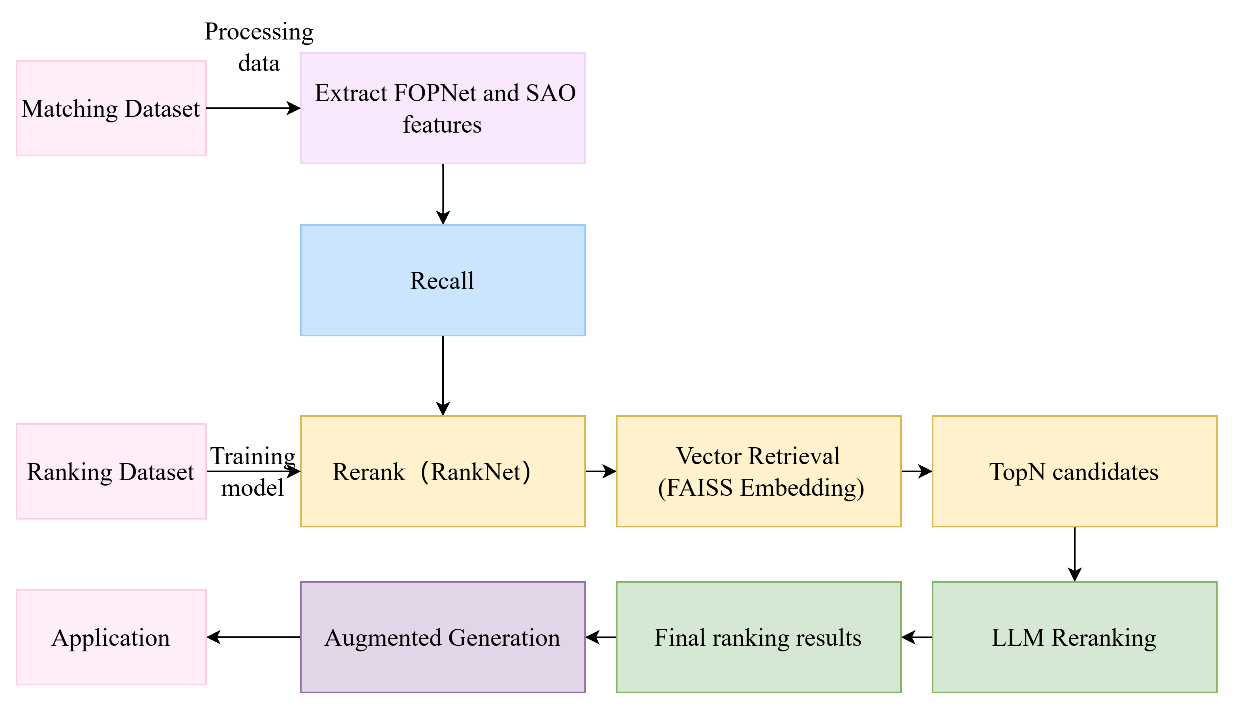


**Figure 7. FOPNet constructed based on examples.**

## Retrieval Application and Validation

Patent retrieval is a complex task in the business world. The technical and domain-specific nature of patent claims makes it challenging for standard keyword search methods to get accurate results. Thus, one of the most important methods for improving the accuracy and relevance of patent database searches is the semantic retrieval. In this section, the performance of FOPNet-based semantic retrieval applications has been addressed to improve understanding of technical information of patents.

The experimental procedure is systematically illustrated in Figure 8, which explains the stages involved in creating and assessing a patent retrieval framework leveraging FOPNet and SAO characteristics. To ensure a full review, a suitably large and diversified patent dataset was prepared in advance. This dataset was gathered from the United States Patent and Trademark Office (USPTO) public database, which allows access to millions of patents via its API. After restructuring, hundreds of semantically comparable patent pairs, particularly those implicated in infringement proceedings, were selected to generate the matching and rating datasets. The procedure begins with feature extraction, where FOPNet triplets and SAO features are extracted from the generated patent datasets. To capture the semantic essence of these entities, FOPNet triplets are embedded into a continuous vector space using FAISS embeddings, enabling efficient retrieval by matrix computations. During the recall phase, the matching dataset is leveraged to determine the most effective matching algorithms, ensuring high-quality candidate retrieval. Subsequently, the obtained results are put through a reranking stage using the learned RankNet model. This step is crucial for refining the search results by putting the most relevant patents at the top of the list. Performance is evaluated using measures such as NDCG (normalized discounted cumulative gain), MAP (mean average accuracy), and MRR (mean reciprocal rank), with particular focus to the ranking position of the desired innovation among the top 100 candidate patents. To further boost interpretability and semantic analysis, large-scale language models (e.g., GPT-4o) are applied. These models give semantic similarity analyses and interpretative insights for the retrieved patents, ensuring robustness in practical implementations. The entire procedure, from feature extraction to final rating and semantic interpretation, offers a comprehensive framework to analyze the retrieval performance of FOPNet in detail. This structured approach, as represented in Figure 8, includes advanced techniques such as vector-based retrieval, machine learning reranking, and large-scale language model interpretation. It provides a scalable and effective approach for semantic retrieval in patent analysis, ensuring both efficiency and accuracy in discovering relevant patents.



**Figure 8. Flowchart of Semantic Retrieval Application**

### Recall and Reranking

Similarity scores for paired patents from the matching dataset were calculated using twelve different similarity calculation algorithms based on FOPNet and SAO features over the recall period, and the results are provided with the average target patent's ranking position among the top 100 patents (i.e. Hit@100) (Teng et al., 2024). Table 6 contains the algorithms based on 12 similarity calculation methods Hit@100 As a result, these algorithms are classified into character level methods, vector level methods, and ontology support methods, which are correspondingly used to the comparative analysis of FOPNet and SAO feature extraction models. From the data in the table, it can be shown that the FOPNet model performs better than standard SAO approaches in most algorithms, notably in ontology assisted similarity algorithms.

Character based methods: Dice, Inclusion, and Jaccard algorithms, the SAO technique Hit@100 They are 62.2, 56.3, and 64.05 respectively, all much higher than FOPNet's 46.9, 45.1, and 46.95.This implies that character level methods are more suitable for addressing the more intuitive text aspects in SAO models, while FOPNet performs rather weakly under this method.

Vector based methods:Under the Euclidean, Pearson, Cosine, and Spearman algorithms, FOPNet performs somewhat worse than SAO. Among them, FOPNet performs under the Euclidean and Spearman algorithms Hit@100 They are 20.15 and 15.8 respectively, while SAO is 26.7 and 28.3 respectively.However, the overall performance of these methods is not as excellent as ontology backed methods, demonstrating that vector level methods have limited usefulness in capturing complicated semantic associations.

Ontology based methods:The ontology support strategy dramatically increased the performance of FOPNet. Under the Lin, Resnik, Jiang, Wu, and Leacock algorithms, FOPNet Hit@100 They are 31, 25.45, 25.05, 34.4, and 45.05 correspondingly, with Wu and Leacock algorithms performing particularly well.In contrast, although the SAO model offers benefits in particular algorithms (such as Wu's 52.4) under these algorithms, its overall performance is not as reliable as FOPNet.This suggests that FOPNet can better identify semantically related patents with diverse text descriptions by introducing functional pattern triplets and ontology backed similarity computations, hence attaining improved retrieval accuracy.

In addition, from the perspective of program execution time, the computation time of character level and word expansion methods is much higher than that of vector retrieval methods. Therefore, in actual applications, vector retrieval is recommended as the ideal method due to its great efficiency, especially in scenarios that necessitate fast response. Combining the data in the table, vector retrieval not only enhances retrieval performance, but also offers a firm platform for subsequent sorting and increased retrieval.

**Table 6 Hit@100 results using diffrerent similarity methods**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Method Types** | **Similarity Algorithms** | | | | | | |
| **Character-based Methods** | **Features** | **Dice** | **Inclusion** | | **Jaccard** |  |  |
| FOPNet | 46.9 | 45.1 | | 46.95 |  |  |
| SAO | 62.2 | 56.3 | | 64.05 |  |  |
| **Vector-based Methods** | **Features** | **Euclidean** | **Pearson** | | **Cosine** | **Spearman** |  |
| FOPNet | 20.15 | | 29.8 | 29.8 | 15.8 |  |
| SAO | 26.7 | 48.2 | | 48.3 | 28.3 |  |
| **Ontology-based methods** | **Features** | **Lin** | **Resnik** | | **Jiang** | **Wu** | **Leacock** |
| FOPNet | 31 | 25.45 | | 25.05 | 34.4 | 45.05 |
| SAO | 32.3 | 33.0 | | 11.1 | 52.4 | 49.3 |

Four vector retrieval approaches were selected for additional testing due to their high recall effect and high efficiency. These vector retrieval recall datasets were subjected to reranking tests using the RankNet model. Three important assessment metrics—MAP (Mean Precision), MRR (Mean Inverse Rank), and NDCG (Normalized Discounted Cumulative Gain)—are compared amongst reordering techniques employing SAO and FOPNet features in Figure 9. The findings show that the FOPNet-based model consistently performs better than the SAO-based model across all measures, irrespective of the similarity retrieval technique (Euclidean, Pearson, Spearman, or Akos). First off, FOPNet's MAP score is noticeably greater than SAO's, suggesting that it is more accurate at locating pertinent patents. For instance, FOPNet clearly outperforms SAO in terms of ranking pertinent patents first because of Pearson similarity. Additionally, the MRR results supported FOPNet's superior ranking accuracy. FOPNet is more successful in placing the most pertinent patents at the front of the ranking than any other similarity approach, as seen by its higher MRR values. Lastly, the most notable distinction between FOPNet and SAO was demonstrated by the NDCG value, which explained the correlation of the ranking elements. FOPNet's ability to better capture patent technology semantics and hierarchical linkages is demonstrated by its persistent superior NDCG scores, especially in Pearson and Arccos similarity.Overall, the rearranged results confirm the efficacy of the knowledge representation backed by the FOPNet ontology, which provides richer semantic understanding through the use of ontology connections and function oriented pattern triplets. In reordering tasks, this puts FOPNet ahead of conventional SAO-based models, which is in line with the research objective of creating a potent patent knowledge graph for more precise patent retrieval and similarity evaluation.

**Figure 9. Comparison of reranking methods using SAO and FOPNet features.**

### LLM-Augmented Generation

While macro-level text similarity evaluations in patent search and analysis can be provided by traditional Large Language Models (LLMs), their analysis usually lacks specificity and quantifiability. In order to more precisely evaluate the technical characteristics of patents, we created the FOPNet triplet extraction method and combined it with the analytical abilities of LLMs to do in-depth comparisons between patent pairs. Large Language Models (LLMs) are added to increase retrieval accuracy and validate the value of FOPNet while also improving retrieval performance. Utilize the FAISS database's integrated vector retrieval mechanism, the top 100 most similar documents that FAISS returned should then be used as prompts to enter into a large model (GPT-4o) for deep similarity analysis.

The large-scale model integrates the FOPNet information of candidate patents and performs a thorough examination of them as a query generator. The documents will then be rearranged to yield more precise search results. To better grasp the FOPNet properties amongst patent documents, the reordering procedure makes advantage of LLM reasoning ability. More accurate ranking optimization is made possible as a result. To guarantee that the most pertinent patent literature appears in the highest position, the ranking has been optimized through numerous optimization rounds. Targeted patent rankings among the top 100 results (Hit@100) are shown in Figure 10 utilizing a variety of techniques, such as RankNet, FAISS Vector Index, and GPT-4o, applied to both FOPNet and SAO models. The outcomes unequivocally show that the FOPNet-based model routinely outperforms the SAO-based method in terms of ranks. FOPNet's average ranking at the initial recall stage is 48.52, which is somewhat higher than SAO's 51.33. FOPNet further improves to 45.51 after reranking with RankNet, however SAO only manages to reach 49.96. Both models are greatly improved by the use of the FAISS Vector Index; FOPNet achieves an amazing value of 8.85, while SAO's ranking is 18.11. Lastly, FOPNet outperforms SAO by a significant margin, achieving the best rating of 5.11 when GPT-4o is utilized for reranking. The ontology-supported knowledge representation of FOPNet, which offers improved semantic comprehension and more precise rankings during the patent retrieval process, is validated by these findings.

**Figure 10. Rankings of targeted patents among Hit@100 results.**

Two very similar patent documents were retrieved from the FAISS patent database: patent A(Path optimization, usage, reinforcement learning algorithms),(Autonomous vehicles, using sensor fusion technology),(Real time traffic conditions, impact, path decision-making process),and patent B(Intelligent navigation system, dependency, machine learning algorithm),(Traffic data, impact, predicted path optimization),(Driving decisions, combined with vehicle historical driving data).We will demonstrate that FOPNet can effectively help models understand deeper semantic similarities between data. Let's first take a look at the similarity analysis report of the FOPNet of two patents by the large model:

|  |
| --- |
| Large Model Analysis Report  **Comparative Requirements:**  patent A：(Path optimization, usage, reinforcement learning algorithms),(Autonomous vehicles, using sensor fusion technology),(Real time traffic conditions, impact, path decision-making process)  patent B：(Intelligent navigation system, dependency, machine learning algorithm),(Traffic data, impact, predicted path optimization),(Driving decisions, combined with vehicle historical driving data)  **Similarity analysis:**  1) Core Functions  Both patents focus on path planning optimization for autonomous driving.  In Patent A, "uses reinforcement learning algorithms," while in Patent B, "relies on machine learning algorithms" is used—both semantically referring to AI-driven path optimization.  2) Object and Data Processing  Both mention real-time traffic conditions influencing the path decision process.  Patent A emphasizes sensor fusion technology, while Patent B relies more on historical driving data.  3) Objectives and Scenarios  Both solutions are designed for vehicle navigation, though Patent A is specific to autonomous vehicles, whereas Patent B applies to broader intelligent navigation systems.  Both emphasize real-time decision-making, but A focuses on instant adjustments, while B includes predictive path planning.  4) Vocabulary and Structure  The phrases "reinforcement learning for path optimization" and "machine learning for route prediction" use different wording but convey similar technical goals.  "Traffic data impacts routing decisions" vs. "historical data influences navigation paths" illustrate complementary perspectives on decision-making inputs.  **Final Remarks:**   * Both patents describe AI-driven path optimization for navigation, with near-identical functional objectives. * Differences lie in data reliance (real-time sensor fusion vs. historical data) and application specificity (autonomous vehicles vs. general navigation systems). * The model determines a high semantic overlap, with a comprehensive similarity score of 0.85. |

Despite sharing similar general objectives and technical domains, FOPNet enhancement study indicates that patents A and B differ significantly in terms of key functional processes, anticipated uses, and implementation technologies. These distinctions are easily discernible at a glance thanks to the ontology architecture of FOPNet, which separates patent data into distinct functional, object, and attribute categories and creates obvious links between concepts.  
In the absence of FOPNet, a sizable language model is typically needed to examine the similarity between two patents by directly evaluating patent data abstracts. This has a number of serious drawbacks. First, there is a large decrease in the efficiency of token usage. When there is a lot of comparative data, a significant percentage of the labels may contain non-essential phrases. This will make it more difficult for the model to identify more pertinent information and produce pointless similarity analysis. Second, it is impossible to comprehend these models structurally. This model just looks at how similar abstract phrases are; it cannot identify or process the relationships between functions, objects, and features in patent applications. Similarity calculations remain superficial due to a lack of knowledge of the technical underpinnings, depending more on lexical and semantic similarity than on substantial improvements in technical capabilities and implementation procedures. By preserving significant and useful structured information and storing patent abstract data as structured triplet data, the FOPNet approach effectively gets over these limitations. Therefore, FOPNet's enhanced LLM analysis offers a more thorough basis for evaluating technological advancements and retrieving patents by supplying structured triplet data and directing in-depth qualitative evaluation of the model:

* Domain-specific concept alignment: Technical terms used in the patent domain are suitably contextualized in accordance with the functional semantic ontology.
* Relationship-conscious analogy: FOPNet not only recognizes entities but also records the important relationships between functions and objects.
* Knowledge representation in a hierarchical structure: From broad concepts to specific implementations, technical comparisons are made easier by FOPNet's ontological framework.

## Conclusions

This study systematically proposes a universal and organized knowledge graph FOPNet, which offers significant support for the patent specification mining to address the semantic challenges. A functional semantic ontology has firstly been constructed with semantically complete entity, relation, extension types employed. In the FOPNet construction stage, the findings have shown that entity and relation extraction results are probably improved with weighted average metrics over 0.8 using pre-trained language models (PTMs). Moreover, knowledge extensions enhancing the conceptual understanding are also integrated into the FOPNet knowledge base. Various knowledge representation learning algorithms contribute a lot to fulfill the FOPNet, and meanwhile FP-Growth and FP-MAX association mining techniques work well for the knowledge correction. Subsequently, the validation stage using the FOPNet methodology has also been fully explored in the semantic retrieval. Performances from the FOPNet have shown more notable gains in recall and reranking periods with higher ranking metrics than those of SAO features. Furthermore, the LLM-augmented generation strategy has also been adopted for the reranking and interpretation, and the results have demonstrated that the FOPNet remarkably improves the Hit@100 metric and provides detailed interpretability with convincing semantic retrieval trials. The effective use of this framework not only validates the capabilities of the FOPNet in the patent community but also acts as a guide for the patent knowledge engineering in a wider variety of technical situations.

## 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

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), 101135. https://doi.org/10.1016/j.joi.2021.101135

Beltagy, I., Lo, K., & Cohan, A. (2019). SCIBERT: A pretrained language model for scientific text. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*. https://doi.org/10.18653/v1/d19-1371

Bergeaud, A., Potiron, Y., & Raimbault, J. (2017). Classifying patents based on their semantic content. *PLoS ONE*. https://doi.org/10.1371/journal.pone.0176310

Chechurin, L. (2016). TRIZ in Science. Reviewing Indexed Publications. *Procedia CIRP*. https://doi.org/10.1016/j.procir.2016.01.182

Chen, L., Xu, S., Zhu, L., Zhang, J., Lei, X., & Yang, G. (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*. 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*. https://doi.org/10.1016/j.eswa.2006.10.015

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*.

Ding, Z., Jiang, S., Ng, F., & Zhu, M. (2017). A new TRIZ-based patent knowledge management system for construction technology innovation. *Journal of Engineering, Design and Technology*. https://doi.org/10.1108/JEDT-03-2016-0017

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

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

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*. https://doi.org/10.1016/j.techfore.2022.121559

Jiang, S., Hu, J., Magee, C. L., & Luo, J. (2022). Deep Learning for Technical Document Classification. *IEEE Transactions on Engineering Management*, 1–17. https://doi.org/10.1109/TEM.2022.3152216

Jiang, S., Sarica, S., Song, B., Hu, J., & Luo, J. (2022). Patent Data for Engineering Design: A Critical Review and Future Directions. *Journal of Computing and Information Science in Engineering*, *22*(December), 1–48. https://doi.org/10.1115/1.4054802

Jung, K., & Park, H. W. (2015). A semantic (TRIZ) network analysis of South Korea’s “Open Public Data” policy. *Government Information Quarterly*. https://doi.org/10.1016/j.giq.2015.03.006

Kim, S., & Yoon, B. (2021). Patent infringement analysis using a text mining technique based on SAO structure. *Computers in Industry*, *125*, 103379. https://doi.org/10.1016/j.compind.2020.103379

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

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*, *8*, 66538–66552. https://doi.org/10.1109/ACCESS.2020.2984024

Li, Z., Atherton, M., & Harrison, D. (2014). Identifying patent conflicts: TRIZ-Led Patent Mapping. *World Patent Information*. https://doi.org/10.1016/j.wpi.2014.07.002

Lin, W., Liu, X., & Xiao, R. (2022). Research on Product Core Component Acquisition Based on Patent Semantic Network. *Entropy*, *24*(4). https://doi.org/10.3390/e24040549

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

Ma, X., & Hovy, E. (2016). End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, *2*. https://doi.org/10.18653/v1/p16-1101

Niemann, H., Moehrle, M. G., & Frischkorn, J. (2017). Use of a new patent text-mining and visualization method for identifying patenting patterns over time: Concept, method and test application. *Technological Forecasting and Social Change*. https://doi.org/10.1016/j.techfore.2016.10.004

Park, H., Kim, K., Choi, S., & Yoon, J. (2013). 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. (2013). Identification of promising patents for technology transfers using TRIZ evolution trends. *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2012.08.008

Park, I., & Yoon, B. (2014). A semantic analysis approach for identifying patent infringement based on a product-patent map. *Technology Analysis and Strategic Management*, *26*(8), 855–874. https://doi.org/10.1080/09537325.2014.909926

Phyo, S. S. (2022). Content analysis-based documentation and exploration of research articles. *Data Technologies and Applications*, *56*(1), 24–43. https://doi.org/10.1108/DTA-07-2020-0146

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, *21*.

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

Spreafico, C., & Russo, D. (2016). TRIZ Industrial Case Studies: A Critical Survey. *Procedia CIRP*. 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*, 103749. https://doi.org/10.1016/j.compind.2022.103749

Teng, H., Wang, N., Zhao, H., Hu, Y., & Jin, H. (2024). Enhancing semantic text similarity with functional semantic knowledge (FOP) in patents. *Journal of Informetrics*, *18*(1), 101467. https://doi.org/https://doi.org/10.1016/j.joi.2023.101467

Wang, F., & Lin, L. (2017). Exploiting Semantic Knowledge Base for Patent Retrieval. In Y. Liu, L. Zhao, G. Cai, G. Xiao, K. L. Li, & L. Wang (Eds.), *2017 13TH INTERNATIONAL CONFERENCE ON NATURAL COMPUTATION, FUZZY SYSTEMS AND KNOWLEDGE DISCOVERY (ICNC-FSKD)* (Issues 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD) CL-Guilin, PEOPLES R CHINA, pp. 2195–2200). https://doi.org/10.1109/FSKD.2017.8393111

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). https://doi.org/10.1007/s11192-019-03191-z

Wodehouse, A., Vasantha, G., Corney, J., Maclachlan, R., & Jagadeesan, A. (2017). The generation of problem-focussed patent clusters: A comparative analysis of crowd intelligence with algorithmic and expert approaches. *Design Science*, *3*, 1–31. https://doi.org/10.1017/dsj.2017.19

Wu, Y., Zhou, F., & Kong, J. (2020). Innovative design approach for product design based on TRIZ, AD, fuzzy and Grey relational analysis. *Computers and Industrial Engineering*, *140*(January), 106276. https://doi.org/10.1016/j.cie.2020.106276

Yan, W., Zanni-Merk, C., Cavallucci, D., & Collet, P. (2014). An ontology-based approach for inventive problem solving. *Engineering Applications of Artificial Intelligence*, *27*, 175–190. https://doi.org/10.1016/j.engappai.2013.07.005

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

Yang, X., Wu, C., Nenadic, G., Wang, W., & Lu, K. (2021). Mining a stroke knowledge graph from literature. *BMC Bioinformatics*, *22*(10), 1–18. https://doi.org/10.1186/s12859-021-04292-4

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

Zhai, D., Zhai, L., Li, M., He, X., Xu, S., & Wang, F. (2022). Patent representation learning with a novel design of patent ontology: Case study on PEM patents. *Technological Forecasting and Social Change*, *183*(100), 121912. https://doi.org/10.1016/j.techfore.2022.121912

Zheng, J., Zhao, Z. yun, Zhang, X., Chen, D. zen, Huang, M. hsuan, Lei, X. ping, Zhang, Z. yu, Zhao, Y. hua, & Liu, R. sheng. (2011). Industry evolution and key technologies in China based on patent analysis. *Scientometrics*, *87*(1), 175–188. https://doi.org/10.1007/s11192-010-0316-3

Zhu, C., & Motohashi, K. (2022). Identifying the technology convergence using patent text information: A graph convolutional networks (GCN)-based approach. *Technological Forecasting and Social Change*, *176*(January), 121477. https://doi.org/10.1016/j.techfore.2022.121477

1. https://www.uspto.gov/patents/ptab/about-ptab [↑](#footnote-ref-1)
2. <https://labelstud.io/> [↑](#footnote-ref-2)
3. https://github.com/zjukg/NeuralKG [↑](#footnote-ref-3)
4. https://rasbt.github.io/mlxtend/user\_guide/frequent\_patterns/fpgrowth/ [↑](#footnote-ref-4)
5. https://rasbt.github.io/mlxtend/user\_guide/frequent\_patterns/fpmax/ [↑](#footnote-ref-5)
6. https://github.com/zjukg/NeuralKG [↑](#footnote-ref-6)
7. https://github.com/MIRALab-USTC/KGE-HAKE [↑](#footnote-ref-7)