ESEA (East-and-Southeast-Asian) Traditional Music Knowledge Base and its Ontology-subgraph-driven NLQ2SPARQL Intelligent Question-Answering System Research[1]

Junjun Cao (Preferred name: Shaojun Si)

Library of China Conservatory of Music, China Conservatory of Music, Beijing, China / Distributed Digital Music Archive and Library Laboratory of Music Technology Area, Schulich Music School, McGill University, Montreal, Quebec Province, Canada

alienmusedh@gmail.com (WeChat Identity: MuseDH; TEL: +86 15011533549, +1 438 596 0579)

Xiaodong Fu

Dean of Library of China Conservatory of Music & Director of the Academic Affairs Office, China Conservatory of Music

Qianping Peng

General Manager/CTO, AgentBoosty Technology Co. Ltd., Hangzhou, China

[1] 本论文为中国音乐学院校内科研专项“基于中华传统音乐馆藏资源的‘知识库’建设和‘知识地图’研究”资助成果，项目编号为 20232005。[Translation: This paper is a research outcome funded by the China Conservatory of Music's internal research project, “'Knowledge Base' Construction and 'Knowledge Map' Research Based on Traditional Chinese Music Collection Resources” (Project No. 20232005)]

ESEA (East-and-Southeast-Asian) Traditional Music Knowledge Base and its Ontology-subgraph-driven NLQ2SPARQL Intelligent Question-Answering System Research

Abstract

We introduce an East-and-Southeast Asian Traditional Music Knowledge Base (ESEA_TM), which, along with the LinkedMusic project, is based on linked data and aims to enhance access across various music databases via AI. ESEA_TM was initially built upon ontology engineering, focusing on cataloging and classification of controlled vocabularies and semantic queries, for example, featuring music classification by music type (music species), and highlighting the use of LLMs to extract traditional instrument entries. We demonstrate the general ontology structure and semantic query examples for ESEA_TM using SPARQL, and especially queries based on knowledge inference. Moreover, in terms of converting Natural Language Questions to SPARQL (NLQ2SPARQL), we provided OWL snippets to LLMs for SPARQL generation. The issue is that, for an oversized ontology, LLMs struggle to pinpoint the corresponding OWL snippet—an ontology subgraph—for correct generation. To address this, we introduce a "subgraph extraction from ontology" approach. The workflow (in Python script) is: (1) Specific Ontology Editing, (2) Ontology Segmentation, (3) Entity Extraction from the Ontology Segments, (4) Subgraph Assembly, (5) SPARQL Generation and Verification Based on Subgraph, (6) Retrieval Augmented Generation (RAG) and Recommendation Based on SPARQL Syntax, (7) Retrieval Recommendation Based on the Neighborhood in Ontology Subgraph. Finally, with cases testing summary, we assess the feasibility, discuss improvement direction and reflect upon a comparison among different orientations for NLQ2SPARQL.

Keywords: large language models (LLMs); shapes; linked data; retrieval augmented generation (RAG); recommendation system; Chinese traditional music; digital humanities; prompt engineering

1 Introduction

There is a growing need for diverse cultural heritage knowledge bases, particularly those based on open and interlinked data, adhering to the FAIR principles (Findable, Accessible, Interoperable, Reusable). Among all kinds of arts heritages, music is particularly prominent. In response to the advocacy of "linked data" (using RDF, Resource Description Framework, as the underlying data model) technology in digital library development, an ongoing project called LinkedMusic: Interlinking Music Resources for Enhanced Access (https://linkedmusic.ca/) aims at connecting various music culture databases globally.[2] However, in terms of diversity, The East-and-Southeast-Asian (ESEA) traditional music culture has been underrepresented. To address this, we introduce the ESEA Traditional Music (ESEA_TM) knowledge base, expanded from the original Chinese Traditional Knowledge Base (CTM) initially built at the Library of China Conservatory of Music, in collaboration with the LinkedMusic community. LinkedMusic's ambition is to facilitate music users in querying the "linked datasets". **However, most (music) users may not be familiar with the linked data's specific query language--SPARQL. To bridge this gap, we have been exploring an approach that prompts the LLMs (Large Language Models) to convert varied Natural Language Questions to SPARQL (NLQ2SPARQL). This approach is specifically intended for "linked database with ontology", which is typical of a knowledge base or knowledge graph.** Hence, our initial experiments employ ESEA_TM, a representative knowledge base, rather than merely a linked database containing only RDF instance data. Retrieval based on such ontological knowledge base can more clearly reflect the organizational logic of knowledge, its structural framework, and flexibly capture knowledge interconnections. In this context, ontologies play a crucial role, not only in domain knowledge definition, cataloging and classification, but also for serving as a reasoning framework, as a "schema" for linked data extraction, supplementation via inference. **Most importantly, we aim to explore and demonstrate, how LLMs can be leveraged with ontology-driven prompt**

[2] See 2 repositories: https://github.com/DDMAL/linkedmusic-datalake and https://github.com/DDMAL/linkedmusic-queries. The latter involves more resources for this article.

engineering to achieve NLQ2SPARQL (including ontology-based SPARQL recommendation) in an applicable workflow.

The first half of the paper introduces ESTA_TM; the second half demonstrates the AI application in such an ontology-driven ESEA_TM knowledge base, highlighting a specific NLQ2SPARQL workflow, along with further evaluation and discussion. The core feature of such workflow is identifying an ontology subgraph that corresponds to a given NLQ and providing it to LLMs, enabling more accurate SPARQL generation.

2 Background of ESEA_TM

The ESEA_TM website[3] is currently in mandarin. Users can conduct various types of queries, including general, advanced, professional queries, knowledge network visualization query (graph query) and most importantly, semantic queries[4], specifically SPARQL query. The knowledge base is more of a metadata base, featuring an ontology-based organization of controlled vocabularies and a knowledge network. The ontology is displayed in both a vocabulary list form and an interoperable graph visualization (using the WebVOWL 1.1.7 tool), accessible through the main page navigation labeled "metadata ontology (元数据本体)". Referring to these ontology vocabulary elements, users can write SPARQL commands on the query interface, where other query examples are also illustrated for reference and the query results are downloadable. Notably, most SPARQL query results are represented as URLs, accessible through a Linked Data Publishing Platform, where each entity with a URL has a public web page, similar to Wikidata pages.

2.1 Issues and Related Work

Many music-related metadata and ontology projects already exist (which we refer to as metadata ontology for short), however, regarding existing issues, most of them are industry-oriented rather than academia-oriented, such as "The Music Ontology" and

---

[3] http://www.usources.cn:8080/dcmusic/home. On the header, navigate along with "知识图谱 > SPARQL 查询" to enter the SPARQL query interface. This website may be shifted in the future. Please refer to the notification on https://github.com/candlecao/Chinese-Traditional-Music-Culture-Knowledge-Base_Cooperation.

[4] http://www.usources.cn:8080/dcmusic/sparql. This knowledge base is supported by Open Link Virtuoso database software, and the SPARQL endpoint is http://www.usources.cn:8891/sparql, on which the user input https://lib.ccmusic.edu.cn/graph/music in Default Data Set Name (Graph IRI) before conducting SPARQL query.

MusicBrainz; or stem from more universal metadata program such as DC, MODS, BibFrame, CIDOC-CRM. Further more, they somewhat exhibit Western-centric bias in the element designing. Here are 2 examples: (1) In music classification, music genre is a dominantly used though very ambiguous concept in academic contexts. In ESEA_TM traditional music classification discourse system, instead, we use the term MusicType (equivalent to "MusicSpecies," a concept entity from the perspective of anthropology of music) instead of music genre. Further, we reinforce this classification with a strictly defined hierarchical framework, formalized using rdfs:subClassOf and skos:broaderTransitive, resulting in a complete directed acyclic graph structure typical of a thesaurus. All of the above have been fully discussed in a previous paper *Research on Construction of "Metadata Ontology" of Chinese Traditional Music Knowledge Base* (Junjun Cao, 2022). (2) Music Species (ctm[5]:MusicSpecies) classification actually depends on traditional music instruments. Regarding instrument classification and cataloging, beyond the renowned tree-hierarchical Hornbostel and Curt Sachs Acoustic Classification,[6] other approaches are relatively flat. Inspired by the core "Work(concept)-Instance-Item" structure of the prestigious BibFrame model, we propose a vertical (rather than flat) three-layer representation for traditional music instrument classification: Instrument concepts/works (anthropological classification viewpoint); instrument instances ("standardized" and acoustic viewpoint); and instrument items (individual/unique physical instruments with recordable acoustic or other characteristic attributes. These 3 layers reflect a transition in viewpoint from the abstract to concrete, interconnected vertically by bf:hasInstance, bf:hasItem, similar to the BibFrame context.

To create the ESEA instrument controlled vocabularies (for the concept layer), we primarily leveraged large language models (LLMs, specifically ChatGPT-4o) to extract structured RDF triples from a semi-structured Oriental Instrument Dictionary corpus. The basic procedures are as follows[7]:

(1) Prepare an ontology snippet containing elements to map to for knowledge extraction. Ontology also serves as a schema in this occasion. In the preparation stage,

---

[5] The name space prefix "ctm" is short for Chinese Traditional Music. Since the ESEA_TM is expanded based on the Chinese Traditional Music, the ctm prefix remains in use.

[6] Refer to https://vocabulary.mimo-international.com/HornbostelAndSachs/en.

[7] To obtain detailed python script and data, see https://github.com/DDMAL/linkedmusic-datalake.

the corresponding classes and properties are chosen, along with their rdfs:label and rdfs:comment. For properties, it's crucial to manifest rdfs:domain and rdfs:range. This process aims to inform LLMs of the knowledge organization structure to be used as a reference for extracting knowledge triples from unstructured corpora.

(2) Example corpora and corresponding extraction results are provided to LLMs. The content from Step (1), combined with detailed example corpora and their correctly extracted triples, serves as the few-shots training context, known as a "prompt." The example corpora include diverse cases, clear, consistent, and comprehensive.

(3) LLMs are prompted to extract RDF triples, with manual verification and iteration . Then, by invoking the ChatGPT API and executing a python script workflow, more extensive corpora are provided to LLMs to extract and output triples.

2.2 Ontology and (SPARQL) Query Examples

Apart from the complete ESEA_TM ontology on the website, Figure 1 presents its simplified graph-visualized overview, showing only representative entities:

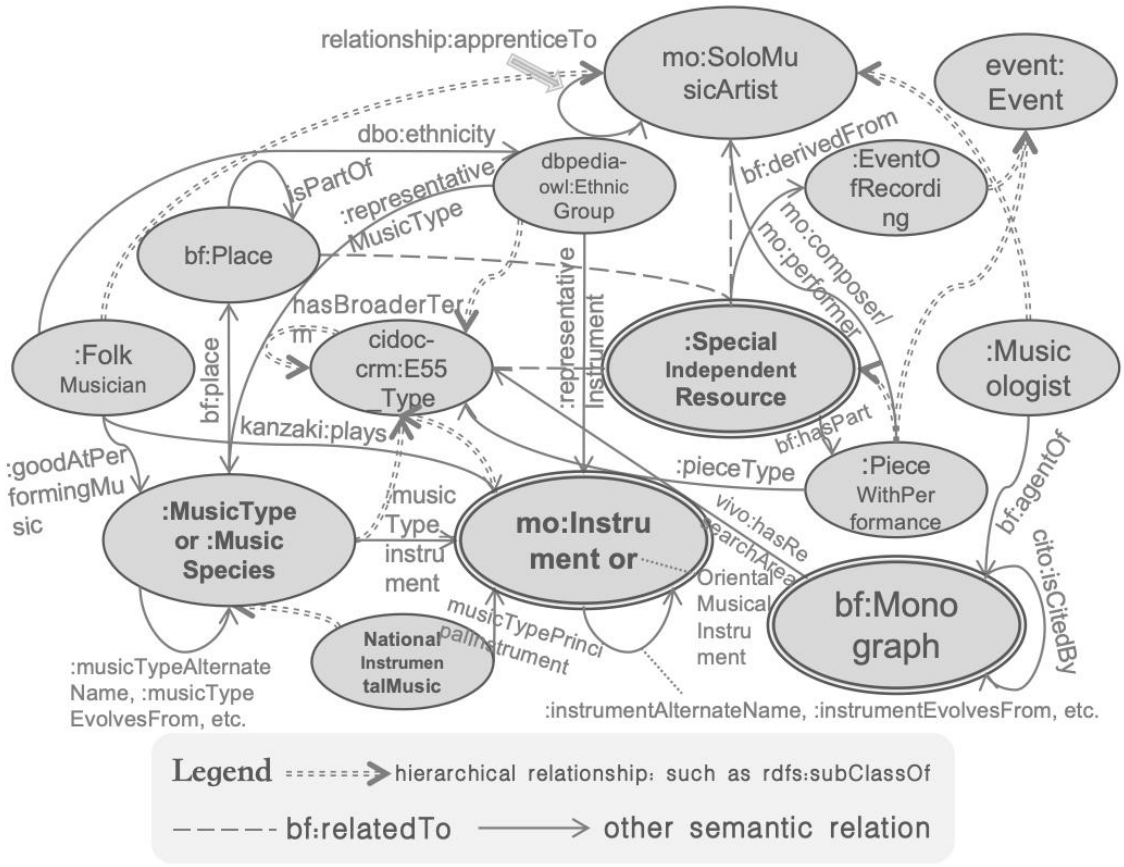

Figure 1. Simplified Overview of ESEA_TM Ontology, Graph-visualization (including only object properties. Junjun Cao, 2022)

As illustrated in Figure 1, ellipses represent classes. Double-edged ellipses signify presence of rich data resources linked to the metadata. One characteristic of the ontology is, the class cidoc-crm:E55_Type bridges macro-ontological elements such as bf:Place, mo:SoloMusicArtist, bf:Work, and micro-ontological elements such as controlled vocabularies or thesauri, including ctm:MusicType, mo:Instrument (or its child-class ctm:OrientalMusicalInstrument), and dbpedia-owl:EthnicGroup. The semantic relationships among the classes are categorized as: (1) parent-child hierarchical relationships, such as rdfs:subClassOf and cidoc-crm:P127_has_broader_term (equivalent to skos:broaderTransitive); (2) bf:relatedTo for a concrete resource, including child-properties such as ctm:relatesMusician, ctm:relatesPlace, ctm:relatesEthnicGroup, ctm:relatesInstrument, and ctm:relatesMusicType; (3) Other normal relationships (labeled alongside the arrows in the figure). We prepared 3 versions of Ontology (rendered in OWL)[8]: (1) The most complete version, available on the website front-end interface; (2) An ontology for knowledge inference; (3) A simplified & specially adjusted ontology (see Section 3.2.1) for the subsequent NLQ2SPARQL processing.

In addition to the examples on the web pages, we prepare other typical pairs of NLQ and corresponding SPARQL queries. These questions are primarily based on a random selection of neighbouring connected classes in Figure 1. The question selection is distributed evenly across the ontology graph to make the examples more representative

[8] Refer to https://github.com/DDMAL/linkedmusic-queries/tree/main/ChineseTraditionalMusicKnowledgeBase/3versionsOfOntology. The 3 ontology versions are named having distinguishable key words respectively as "EasternMusic""withAdditionalAnnotation""withAdditionalAnnotation...simplified" corresponding to the order of (1)(2)(3) of the main text. Their graph URI are respectively (1) http://ont.library.sh.cn/graph/ctm_1.1, (2) https://lib.ccmusic.edu.cn/graph/music_inf, (3) http://ont.library.sh.cn/graph/ccmusic_lit_inf. Users can retrieve the ontology graph on the SPARQL endpoint: http://www.usources.cn:8891/sparql by filling out the Default Data Set Name (Graph IRI) with the corresponding URIs of (1)(2)(3).

and comprehensive. Readers can refer to GitHub to see the dynamically updated question selection.[9] For example:

(1) EthnicGroup -> MusicType -> Place: 蒙古族有哪些音乐类型_乐种，这些乐种的分布地域各是什么？(Translation: What music types exist in the Mongol ethnic group, and what are the regional distributions of these types?)

Answer, e.g.: music type 乌力格尔(Uliger), located in 内蒙古自治区 (Inner Mongolia Autonomous Region).

(2) Instrument -> EthnicGroup: 鼻箫这件乐器是哪个民族的？潮尔又是哪个民族的？(Translation: Which ethnic group does the nose flute belong to? And which ethnic group does Tsuur belong to?)

Answer, e.g.: The nose flute is of 黎族(Li ethnic group), 高山族(Gaoshan ethnic group); Tsuur of 蒙古族(Mongolian).

(3) MusicType -> EthnicGroup: 喊傣亮、四簧口弦琴曲、御神歌、“宫廷筵宴乐-缅甸乐”、加美兰、扎平各是哪些民族的？[Translation: Which ethnic groups do Hăn Dăi Liàng, four-reed jaw harp music, Sacred Shinto Hymns(ぎょしんか), “Imperial Banquet Music – Burmese Music”, Gamelan, and Zh ā píng belong to?]

Answer, e.g.: 喊傣亮 is of Dai ethnic group; 四簧口弦琴曲 is of Gaoshan ethnic group and Atayal ethnic group; 御神歌 is of Ryukyuan people; 宫廷筵宴乐-缅甸乐 is of Myanmar; 加美兰 is of Javanese; 扎平 is of Malays.

(4) MusicType -> SpecialIndependentResource -> Place: 我馆有什么特藏资源涉及甘美兰音乐，该资源涉及的地域在哪？(Translation: What special collection resources does our library have related to Gamelan music, and which places do these resources cover?)

[9] https://github.com/DDMAL/linkedmusic-queries/tree/main/ChineseTraditionalMusicKnowledgeBase/NLQ2SPARQLworkflow, where please refer to 2 folders: (1) sampleQuestions (2) storageOfPairsOfNLQ&correspondingCorrectSPARQL. The files are labeled according to the substantial subject-(predicate)-subject chained structure. Find the corresponding SPARQL and execute them on the http://www.usources.cn:8080/dcmusic/sparql endpoint.

Answer, e.g.: a resource having subject “甘美兰（Gamelan）音乐(音乐学系汇报)”[Gamelan Music (Musicology Department Report)] related to 东南亚 (Southeast Asia) and a resource having subject “陈铭道印度尼西亚巴厘岛之行”(Chen Mingdao's Trip to Bali, Indonesia) related to 巴厘岛(Bali)

(5) MusicType -> SpecialIndependentResource: 朝鲜族民歌、潮尔在我馆是否有对应的特藏资源收录？[Translation: Does our library have any special collection resources on Korean folk songs and Tsuur(Chao'er)?]

Answer, e.g.: “俄罗斯图瓦 ALASH 乐队演出（呼麦组合）”[Russian Tuvan ALASH Band Performance (Throat Singing Group)]...

(6) SpecialIndependentResource -> MusicType, Instrument, EthnicGroup: 《越南闲愁》涉及到什么音乐类型(乐种)、乐器、民族？(Translation: What music types, instruments, and ethnic groups are involved in “Vietnamese Idle Sorrow”?)

Answer, e.g.: It involves “民族器乐”(national instrument music) and “歌舞音乐”(singing and dancing music), instruments including “铜锣”(copper gong), “竹筒”(bamboo tube), “月琴”(moon guitar), ethnic group/nation as “越南”(Vietnam).

There are some special Queries:

(1) Query for blank node structure, for example:

A. 伽倻琴在哪个民族中又称作嘎牙高？(가야금) (Translation:In which ethnic group is the gayageum also known as gayagum?; answer: Korean)

```
PREFIX ctm:<https://lib.ccmusic.edu.cn/ontologies/chinese_traditional_music#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
SELECT * where {
?s rdfs:label "伽倻琴" ;
    ctm:instrumentAlternateName ?blankNode .
?blankNode ctm:instrumentAlternateName/rdfs:label "嘎牙高" ;
            ctm:ethnicGroup/rdfs:label ?ethnicGroupLabel . }
```

B.苗族大唢呐在苗族人（苗族语）中又称作什么？[Translation: What do Miao people (in their language) call the large Miao suona/surnay?; answer: 梭拉(suola), 亮巴(laingba), 塞豁(saihuo), 列罗(lieluo]

```
PREFIX ctm:<https://lib.ccmusic.edu.cn/ontologies/chinese_traditional_music#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
SELECT * where {
?s rdfs:label "苗族大唢呐" ;
    ctm:instrumentAlternateName ?blankNode .
?blankNode ctm:instrumentAlternateName/rdfs:label ?instrumentLabel ;
            ctm:ethnicGroup/rdfs:label ?ethnicGroupLabel .}
```

(2) Geographical coordinate query

县级行政单位湖北省荆门市东宝区方圆 60 公里之内有什么乐种？其中，哪些在中国音乐学院图书馆有收藏？[Translation: What types of music types are within a 60-kilometer radius of Dongbao District, Jingmen City, Hubei Province? Among them, which are collected in the library of the China Conservatory of Music?]

```
define input:inference 'urn:owl.ccmusicrules0214'
PREFIX gn: <https://www.geonames.org/ontology#>
PREFIX ctm: <https://lib.ccmusic.edu.cn/ontologies/chinese_traditional_music#>
PREFIX bf: <http://id.loc.gov/ontologies/bibframe/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
select
distinct ?OtherCity ?OtherCounty ?distance ?MusicType2 ?MusicType3 ?SpecialIndependentReso
urce
where { ?CenterCounty rdfs:label "东宝区" ; # Dongbao District
                        wdt:P625 ?centerCoordinateLocation .
  optional { ?MusicType1 bf:place ?CenterCounty ; a ctm:MusicType }
  ?OtherCounty wdt:P625 ?otherCoordinateLocation ;
                gn:parentADM2 ?OtherCity .
  optional { ?MusicType2 a ctm:MusicType ;
                        bf:place ?OtherCounty .
}
```

```
        optional { ?MusicType3 a ctm:MusicType ;
                                bf:place ?OtherCity .
      }
        bind (bif:st_distance(?centerCoordinateLocation, ?otherCoordinateLocation)
AS ?distance)
        filter (?distance<=60)
      } order by ?distance
```

2.2.1 Query Based On Knowledge Inference

(1) Since we have emphasized the hierarchical parent-child relationship of classes in the ontology, as well as parent-child properties, all queries are based on data supplementation using ontology-based inference. In addition, other semantics in the OWL, such as owl:inverseOf, owl:equivalentTo for inference will also be effective. So put a praevia line of SPARQL code `define input:inference 'urn:owl.ccmusicrules0214'` in the front before executing any SPARQL query. (2) The other situation involves rule-based knowledge inference. For example, there is a child class of bf:Place, which is called ctm:ChinaJurisdiction, containing tiers of Province, City, County, Town, Village. Correspondingly, we create rules using SPARQL CONSTRUCT key word, such as: if a Music type is distributed in a village, then it is also distributed in the town that contains the village, the county that contains the town, and so on. Above all, the supplemented data based on knowledge inference lay a foundation for the subsequent NLQ2SPARQL cases.

3 NLQ2SPARQL Development Based on LLMs

Theoretically, an ontology-driven knowledge base is the most ideal one. The ESEA_TM knowledge base is such one, with an ontology engaged, which means, principally, each entity (or instance, to be precise) is explicitly classified with rdf:type, necessitating an RDFS defined simple ontology structure underlying the instance data (or extractable from the instance data), where each property has a clear definition of its rdfs:domain and rdfs:range. in our RDF database, for example:

```
ctm:musicType_instrument rdf:type owl:ObjectProperty ;
                                rdfs:domain ctm:MusicType ;
                                rdfs:range mo:Instrument ;
```

```
                                    rdfs:comment "关联一个音乐类型(乐种)和它使用的乐器（其乐器编制）"@zh , "Associate a music type with the instruments it uses (its instrumental arrangement)"@en ;
                                    rdfs:label "乐种使用…乐器"@zh , "MusicTypeUsesInstrument"@en .
        ctm:MusicType rdf:type owl:Class .
        mo:Instrument rdf:type owl:Class .
```

This "property + class in domain + class in range" structure functions like a schema in a relational database. Our research for NLQ2SPARQL is premised on this or otherwise, inspired by the use of schema in NLQ2SQL research.

3.1 Related Work (Prior Experiment, Issue and Literature Review)

We conducted prior experiments by providing LLMs with specific ontology (such as above) snippet corresponding to a given NLQ[10] such as 福建南音这个乐种会使用到什么乐器？(translated as: What instruments does the music type *Fujian Nanyin* use?), then LLMs (We used chatGPT 4o) could generate a correct SPARQL such as:

```
define input:inference 'urn:owl.ccmusicrules0214' # This row is not generated by GPT
prefix ctm: <https://lib.ccmusic.edu.cn/ontologies/chinese_traditional_music#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
select * where { ?MusicType rdf:type ctm:MusicType ;
                                    rdfs:label "福建南音" ; # labeled "Fujian Nanyin"
                                    ctm:musicType_instrument ?Instrument .
                    ?Instrument rdfs:label ?instrumentLabel }
```

However, given the current capacity of LLMs, this approach has a limitation, causing inconvenience: different NLQs require different ontology snippets. To address this variance, providing the entire ontology to LLMs seems advisable; however, the more the larger ontology, the more LLMs struggle to focus on the correct corresponding snippet for the SPARQL generation.

---

[10] In our LinkedMusic's 2024 yearly conference, we had a report recorded in the file "Leveraging Chat GPT for Natural Language Query to SPARQL on Virtuoso.pdf". (See Github). In such report, for the same query request, we made a comparison between SPARQL query and query on the front end interface of conventional database, showcasing the flexible performance of SPARQL.

In parallel, we have conducted literature reviews, summarized as follows:
On one hand, few previous studies advocate directly using ontology for NLQ2SPARQL; most approaches rely on matching existing RDF data examples or SPARQL query examples. L. Kovriguina (2023) proposed the SPARQLGEN method , a one-shot approach for generating SPARQL queries by prompting LLMs, which involves augmenting LLMs with a knowledge graph fragment, referred to as a subgraph. The subgraph contains typical triples that are required and serve as context for prompting LLMs to generate the correct SPARQL query. Subsequently, Dmitrii Pliukhin (2023, November) developed an improved subgraph extraction algorithm and an improved approach for selecting a guiding example (of a similar NLQ and its corresponding SPARQL query) to enhance the robustness of SPARQL generation. Shuangtao Yang et al. (2023) proposed a method with a "Selected Schema," featuring the extraction of entities from the NLQ, mapping the entities to the graph, selecting relevant properties from entities, and combining these entities as context for SPARQL generation. I hold this method of symbolizing selected schema is essentially no different from the subgraph extraction method described above. These methods are based on the premise that (1) an unknown and particularly large linked database lacks a comprehensive ontology as a schema reference, and (2) subgraph retrieval involves considerable algorithmic complexity, which is difficult for most librarians to master and maintain. Further, neural SPARQL query generation architectures will be considered in future work for literature review and method exploration.

On the other hand, some papers indeed refer to leveraging schema (ontology or shapes) in the process of prompting LLMs for SPARQL generation. Davide Varagnolo et al. (2023) proposed an approach to perform a syntactic analysis of given NLQs -> using DRS (Discourse Representation Structure, also interpreted in an ontology) to achieve an ontology matching: assigning an ontology class to each discourse referent and an Object Property or Data Property to each condition. His work concerns conventional NLQ2SPARQL without LLMs. Dean Allemang (2024) proposed using LLMs and a "Zero-shot Prompt" for SPARQL generation based on NLQ and the Ontology of a Knowledge Graph. "Zero-shot" means not referring to any example pairs of NLQs and their corresponding SPARQL queries, unlike many other papers. He also used ontology to validate the generated SPARQL, calling this approach "ontologies to the rescue" for

increasing the accuracy of LLMs2SPARQL. Regarding validation, Vincent Emonet (2024) proposed using “shapes” to validate SPARQL generation instead of ontology, due to its open-world hypothesis, which is unsuitable for validation. He/she also emphasized leveraging example pairs of NLQ&SPARQL stored in vector database as prompts for LLMs2SPARQL, in which case, user questions are matched to the similar NLQs in the vector database, then the corresponding SPARQL is retrieved as a one-shot training example for the new SPARQL generation.

In our paper, we highlight an LLMs-based NLQ2SPARQL with prompt engineering. Therefore, a key issue is: What is the most appropriate prompt to provide LLMs as context? According to the previous literature reviews, we can summarize 3 different kinds of prompt: (1) subgraph: typical RDF instance data extracted from the general graph, relying on graph traversal and is algorithm & computing-strength-oriented; (2) example pairs of NLQ and corresponding SPARQL stored in advance, relying on calculating the similarity between a new NLQ and an existing sample NLQ; (3) schema: either ontology or shapes, fundamental to the knowledge-representation-oriented/semantic-oriented approach; with only schema as prompt, the work flow can be considered zero-shot learning. While (1)(2) are usually viewed as in association with one-shot or few-shots learning.

Currently, it is not commonly assumed that a graph database (an RDF database can be considered a graph) requires a schema, and most large graphs do not have one (such as Wikidata, which lacks an underlying ontology). This may result in insufficient attention to ontology (or shapes)-driven NLQ2SPARQL process research. In addition, shapes can describe the structure of an extant RDF graph, such as which properties a class currently connects to or can connect. Aligning with this structure, we can create a “nominal” ontology for an RDF graph, provided that each instance has an attributed class. “Nominal” means using rdfs:domain and rdfs:range to describe “which classes a property can connect.” Strictly speaking, this asserts that the Subjects or Objects (S or O) of a given property must fall under a certain class scope, although S or O may be other classes. Vincent Emonet criticized that ontology is based on Open World Hypothesis (as mentioned earlier) while shapes are based on Close World Hypothesis, which is more suitable for validating the SPARQL generated from NLQ by LLMs. However, our

experiment demonstrated that there are workarounds using ontology to simulate effective validation [see Section 3.2.1 > (2) > B and Section 3.2.5].

3.2 Intelligent Question-Answering Workflow

Since our ESEA_TM knowledge base has an aligned ontology with a corresponding inference mechanism activated, we advocated and achieved a preliminary "prompting LLMs for NLQ2SPARQL" workflow that relies solely on ontology, featuring a "subgraph extraction from ontology" approach. In this context, the subgraph differs from the "instance subgraph" discussed previously in the literature review. Instead, it is an "ontology subgraph" extracted from the complete ontology graph, as illustrated by the topological visualization of ontology on our website, where properties are represented as edges while the rdfs:domain and rdfs:range of properties link classes as nodes.

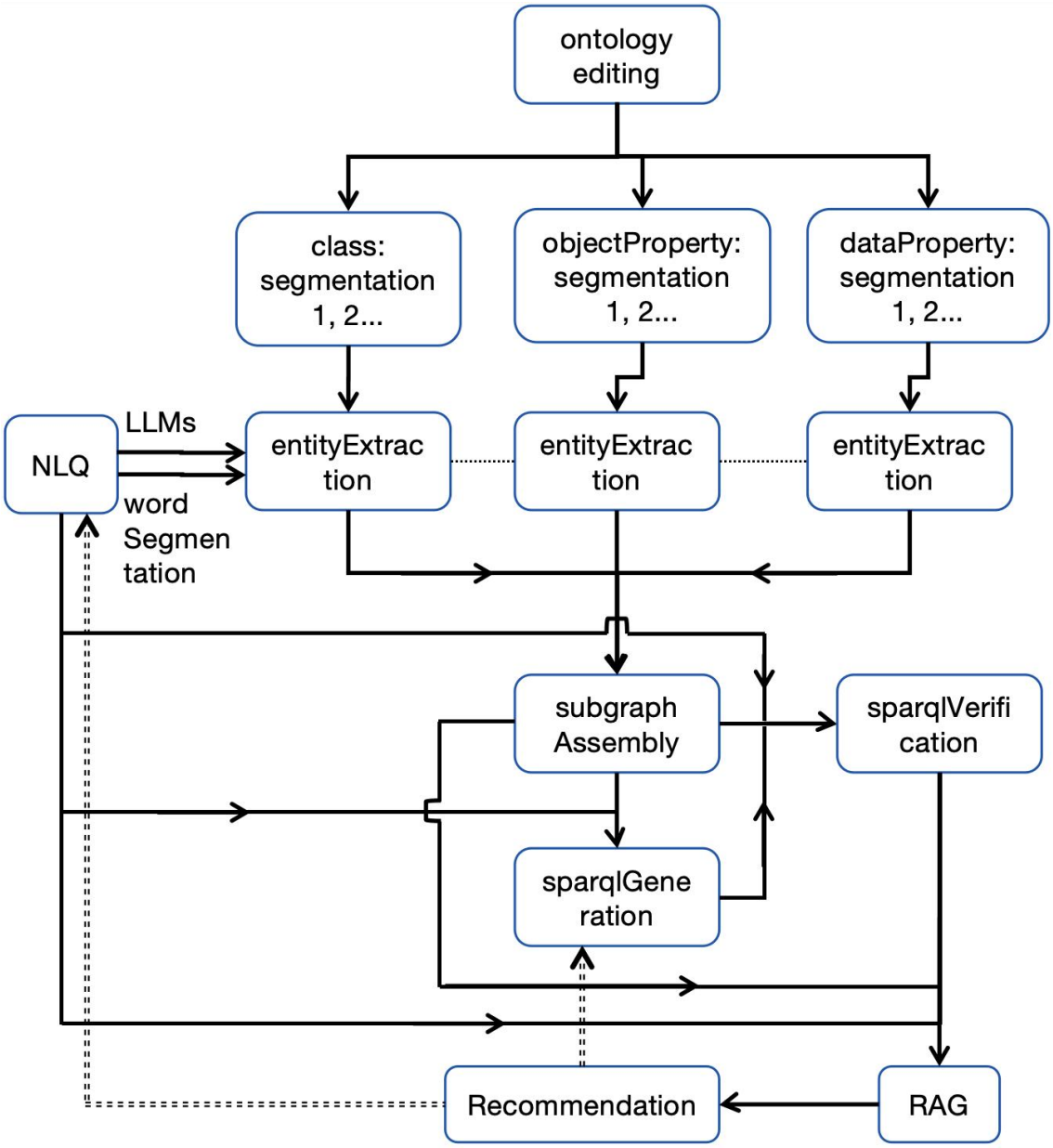

Figure 2. The Ontology-subgraph-driven Intelligent Question-Answering Workflow

Corresponding to Fig.2, the general idea is to segment the specifically edited OWL file into chunks (e.g., class chunks, object property chunks, and data property chunks), each of which is "digestible" in a single-turn interaction with LLMs. Every chunk is considered a list of vocabularies, and LLMs are prompted to extract classes and properties from NLQ by mapping to "entities" in a Turtle OWL file, similar to retrieving entries from a vocabulary dictionary.[11] We call this process "entity

[11] We use an RDF/Turtle-formatted OWL document to represent the ontology. In this file, both classes and properties are referred to as "entities." However, in the context of RDF or knowledge extraction in knowledge graphs, "entity" typically refers to an instance rather than a property or class.

extraction." All the extracted entities are then reassembled to form an ontology subgraph. Based on this subgraph, LLMs can generate SPARQL more accurately for a given NLQ. Then, the NLQ, the generated SPARQL, and the ontology subgraph are gathered for SPARQL verification. After the execution of this SPARQL query, Retrieval Augmented Generation (RAG) is performed. LLMs are prompted to re-illustrate the answer to the NLQ and recommend other SPARQL queries for reference, along with which, corresponding NLQs yield inversely. This ontology-subgraph-driven workflow includes the following detailed steps:

3.2.1 Specific Ontology Editing

We previously introduced preparing distinct ontology versions. This section explains why a specific ontology version is needed, as the existing one is inadequate or excessive. For the subsequent Ontology Segmentation *(3.2.2)*, Entity Extraction... *(3.2.3)* and Subgraph Assembly *(3.2.4)*, etc., specific editing includes: (1) Clarifying or enriching annotations for rdfs:label or rdfs:comment of the entities in a turtle OWL file, such as adding multiple synonymous labels and illustrating in comments how a property connects a class in its domain and a class in its range[12]. To ensure that any property can be retrieved and reassembled *(3.2.4)* into the ontology subgraph, neither its domain nor its range should be left blank. (2) Semantic reinforcement: A. Ensure that every object property has a corresponding inverse property, because in NLQ, the positional relationship between a subject and an object can be reversed, e.g., "What instrument does this music type use?"(S: music type; O: instrument. Please refer to the same NLQ and SPARQL in Section 3.1) can be reversely phased as "What instrument has this representative music type?"(S: instrument; O: music type). Regarding the underlying structure of Subject-Predicate-Object, the inverse property of ctm:musicType_instrument is ctm:instrumentRepresentativeMusicType, the two of which have opposite domain and range and should both be stated in OWL;[13] B. Add constraint assertion, especially negative assertion, serving as a substitute for shapes. For

[12] For this comment illustration, refer to the ontology OWL file containing the name "...withAdditionalAnnotation..." and not containing "simplified" (for the URL, refer to the next footnote).

[13] Refer to the ontology OWL file containing the name "...withAdditionalAnnotation...simplified" on https://github.com/DDMAL/linkedmusic-queries/tree/main/ChineseTraditionalMusicKnowledgeBase/3versionsOfOntology.

example, to specify that the domain/range of a property must not be a certain class, the OWL script should be: “<Property> rdfs:domain/rdfs:range [rdf:type owl:Class; owl:complementOf <SomeClass>]”. C. Currently, we have found that if a domain/range is a certain class, both its parent class (except for owl:Thing) and its child class should be added to the domain/range using a union relation assertion. Two detailed examples are provided in Section 3.2.5. (3) Unnecessary triples are deleted to reduce the token load on LLMs. This step is ongoing and not limited to the initial stage.

3.2.2 Ontology Segmentation

As previously mentioned, LLMs struggle to process an oversized ontology. Otherwise, Ontologies can be viewed as a vocabulary list for entities (broadly, the entities include classes, properties, etc.), where each entity is described by a set of triples. Therefore, the OWL file is segmented into different manageable entity segment, then provided to LLMs, which improves its concentration in processing the ontology segments. Considering LLMs’ token limitations, the segmented OWL file needs only to contain triples composed of the rdfs:label and rdfs:comment properties . Effective segmentation can be achieved by dividing the ontology into 3 initial groups: (1) class entities, (2) object property entities, (3) data property entities. This segmentation enables us to differentiate and refine specific prompts for each group and avoid interference from entities with identical labels (literally overlapping entities). For example, both the object property bf:place (label: "分布地域," meaning "the place where it's distributed") and the class bf:Place share the same label. Separating these into different prompting rounds helps prevent either from being omitted due to this label overlap during the subsequent entity extraction process (Section 3.2.3). The size or number of segmented groups can vary, depending on the ontology’s overall size and the LLMs’ evolving capacity.

Two additional guildelines are helpful: (1) To minimize the risk of missing entities in subsequent steps, it’s beneficial to group similar entities separately, including those in parent-child relationships, such as bf:Place and its child class places:Province. (2) Object properties that are inverse properties of each other had better be separated into different subgroups.

3.2.3 Entity Extraction from the Ontology Segments

LLMs are prompted to extract entities from an NLQ by mapping them against the entities in all segments of the ontology. Here, the "entities" encompass classes, properties, and instances. 3 key points are included: (1) First, without initial reference to ontology, LLMs are prompted to perform entity extraction directly from NLQ, then to conduct word segmentation, especially crucial for unspaced Chinese NLQs. (2) The corresponding entities (including the segmented entities) are then mapped to the entities in the segmented OWL files, where the clarified annotations in Section 3.2.1 serve as effective references. (3) The extracted entities from Point (1) may not be explicit classes but could only be instances, for which, SPARQL is used to retrieve their rdf:type values from the SPARQL endpoint. The retrieved values are then collected together with other extracted entities. Above all, the entity extraction relying on LLMs is not stable sometimes, so the strategy prioritizes comprehensiveness over precision, favoring the inclusion of potentially redundant entities to ensure no relevant information is omitted, which instead, will contribute to an ontology subgraph based retrieval recommendation, as demonstrated in Section 3.2.7.

For example, in the previous question selection (2) of Section 2.2, the NLQ "潮尔又是哪个民族的" (Translation: then which ethnic group does Tsuur belong to) led to the extraction of "潮尔." Instances with rdfs:label value "潮尔" were identified as either a ctm:MusicType or an mo:Instrument. The two retrieved classes are then combined with other explicitly retrieved classes. Additionally, classes such as rdfs:Literal are added to the extracted classes group, given that many rdfs:range values of data properties in our OWL file are classified as rdfs:Literal.

3.2.4 Subgraph Assembly

If entity extraction is the "key" (for robustness) of this approach, then its "distinctive feature" is the subgraph assembly described here. In short, if the entire ontology is considered as a large community, subgraph assembly is like performing community detection based on a given NLQ – identifying small communities related to the NLQ within the larger one, thereby achieving targeted SPARQL generation.

The python script corresponding to Section 3.2.2 and 3.2.3 is named "1_entitiesExtractionFromNLQ_basedOnOntology.py"; the following procedures corresponds to "2_subGraphAssemblyFromOntology_3_SPARQLgeneration.py".[14]

The extracted classes and properties mentioned above are reassembled to form a subgraph of the general ontology graph. This subgraph consists solely of a selection of the extracted classes and properties. A filtering process is applied before the assembly, whose key criterion is to identify "subject-predicate-object" semantic patterns formed by the given classes and properties. For example, if one of the extracted classes appears in the domain of one of the extracted object properties, and another the extracted classes appears in the range of that object property, then both classes and that object property are all returned, along with their descriptive triples. This process yields an assembled subgraph, small enough to be digestible in a single-turn of prompting LLMs. The subgraph may include some unnecessary entities, but this is just not problematic, as they can be used for recommending other related SPARQL patterns.

3.2.5 SPARQL Generation and Verification Based on Subgraph

Based on the subgraph ontology, LLMs (specifically, the "claude-sonnet-4-20250514" model) are prompted to generate the SPARQL query for a given NLQ. The prompt reminds that literal entities in the NLQ should be rendered using rdfs:label, associated with the variable in the SPARQL query. To ensure robustness, LLMs are then prompted for a "reflection" step, using the prompt pattern: "*Given the corresponding ontology snippet and natural language question as context, cross-check the SPARQL query to ensure its syntactic and semantic correctness, to avoid misinterpreting the natural language question... and return the modified SPARQL query if necessary.*" Here, the ontology, rather than shapes, is also used for SPARQL verification. An additional prompt addresses potential NLQ ambiguity: "*If you are uncertain about specific classes, properties, or semantic structure, broaden the retrieval scope using syntax such as: UNION... | operator... OPTIONAL...*" Finally, the code `define input:inference 'urn:owl.ccmusicrules0214'` is added into front of the generated SPARQL script.

[14] See https://github.com/DDMAL/linkedmusic-queries/tree/main/ChineseTraditionalMusicKnowledgeBase/NLQ2SPARQLworkflow.

To testify ontology can substitute for shapes, 2 examples are provided: (1) NLQ:“四川扬琴是否有代表性曲目？”(Translation: Does Sichuan Yangqin have representative pieces?) For this query, the corresponding ontology subgraph has a property ctm:representativePiece to be used for the corresponding SPARQL query. However, when analyzing the semantic structure, LLMs are prone to making its subject (in rdfs:domain) an instance of ctm:PieceWithPerformance, possibly because the semantic meaning, such as “a bunch of music pieces has its representative piece,” also sounds logical. In order to weed out the disturbance of ambiguity, we adjust the OWL as follow:

```
ctm:representativePiece rdfs:domain [ rdf:type owl:Class ;
                                      owl:unionOf ( ctm:MusicType
ctm:OrientalMusicalInstrument )] ,
                                    [ rdf:type owl:Class ;
                                      owl:complementOf ctm:PieceWithPerformance ] .
```

which constrains the domain to exclude ctm:PieceWithPerformance. (2) NLQ:“壮族有哪些音乐类型(乐种)，其中哪些音乐类型(乐种)在云南省？”(Translation: What music types does the Zhuang ethnic group have, and which of these music types are found in Yunnan Province?) Initially, the generated SPARQL query[15] didn’t constrain the type of the variable ?musicType to be ctm:MusicType. Even though the rdfs:range of ctm:representativeMusicType [with the rdfs:label value “民族有…代表性乐种” (Translation: Ethnic group has ... representative music types)] is ctm:MusicType, due to open world hypothesis, the range could still be of other classes, such as mo:instrument. To reinforce a clear constraint, we have added the assertion `ctm:MusicType owl:disjointWith mo:Instrument .`. Therefore, it can be inferred that the range of ctm:representativeMusicType must not be of class mo:Instrument. Subsequently, LLMs reinforce the constraint by asserting `?musicType a ctm:MusicType .`, which makes the query result more confined and accurate.

[15] See EthnicGroup_MusicType_Place1.sparql, from https://github.com/DDMAL/linkedmusic-queries/tree/main/ChineseTraditionalMusicKnowledgeBase/NLQ2SPARQLworkflow/storageOfPairsOfNLQ%26correspondingCorrectSPARQL%26others.

3.2.6 Retrieval Augmented Generation (RAG) and Recommendation Based on SPARQL Syntax

The verified SPARQL query is executed by invoking the corresponding endpoint. LLMs are then prompted to illustrate the returned query result in respect to 3 contexts: (1) the original NLP, (2) the related ontology subgraph, (3) the executed SPARQL query. To elaborate, on one hand, LLMs integrate its own pre-trained knowledge with the retrieved knowledge, so as to find inadequacy or inconsistency in the result, thus enriching the explanation through integration. For example, ESEA_TM has also been attempting reconciliation with Wikidata (, which reconciliation is also a key step in LinkedMusic project). Because many URIs (whether properties or classes) are recognizable by pre-trained LLMs, this reconciliation work will contribute to a unified account of knowledge. On the other hand, LLMs determine: (1) If the result set is excessively large, a statistical analysis followed with a summary is performed. (2) If the result is insufficient or even empty, the retrieval scope is broadened by relaxing query conditions/constraints in the SPARQL query, so alternative query patterns can be recommended.

Through SPARQL syntax modification, constraints can be relaxed to increase result yield. For example, for the question “北京锣鼓乐有哪些子类？” [Translation: What are the sub-types of Beijing Luogu (Gong and Drum) Music?], a direct SPARQL query is:

```
define input:inference 'urn:owl.ccmusicrules0214'
PREFIX ctm: <https://lib.ccmusic.edu.cn/ontologies/chinese_traditional_music#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?subType ?subTypeLabel
WHERE {
  # Identify the Beijing Luogu music type entity
  ?BeijingLuogu a ctm:MusicType ;
                rdfs:label "北京锣鼓乐" .
  # Find all subtypes using the narrower term relationship
  ?BeijingLuogu ctm:musicType_narrowerTerm ?subType .
  # Get the labels of these subtypes
  ?subType rdfs:label ?subTypeLabel . }
```

and the result contains 2 instances: “北京花会锣鼓”[Beijing Huahui(Festive) Luogu], “北京丧事锣鼓”[Beijing Sangshi(Funeral) Luogu]. LLMs interpreted “sub-types” as only child types, excluding grandchild types, great-grandchild types, etc. However, in its subsequent recommendation process, it recommended 3 alternative query plans: A. using `?BeijingLuogu ctm:musicType_narrowerTerm* ?subType .` (adding *[16]), retrieve child types, grandchild types, etc., as all subordinate types; B. using `FILTER(CONTAINS(str(?label), "北京") && CONTAINS(str(?label), "锣鼓"))`, retrieve instances that have both “北京” (Beijing) and “锣鼓” (Gong and Drum) in their label; C. find other instances of `ctm:MusicType` that are directly connected to Beijing Luogu (Gong and Drum) Music, regardless of whatever properties used in the connection. These broadening or associating method allow us to find related instances.

Part of the prompt for verification, illustration or recommendation is excerpted below:

> *...*
>
> *please "broaden the retrieval scope" by relaxing query conditions/constraints in the SPARQL or recommend other potential query patterns.*
>
> *For example:*
>
> *4.1 may use the UNION keyword to include multiple options to interpretate a question, especially when the question can be divided into multiple sub-questions, or in case of handling an objectProperty and a dataProperty which have the similar semantic meanings*
>
> *4.2 use the | operator to represent a logical OR for properties*
>
> *4.3 use the OPTIONAL keyword:*
>
> *4.3.1 also useful when handling an objectProperty and a dataProperty which have the similar semantic meaning, etc.*
>
> *4.3.2 to allow partial matches, ensuring that queries remain valid even when certain properties or property values are missing. It is particularly beneficial for handling uncertain or "if, possibly" relationships (e.g., "Something may relate to something else") or when managing properties with similar semantics*
>
> *...*
>
> *4.4 remove class constraint on a variable to broaden the retrieval scope*
>
> *4.5 cancel FILTER condition to broaden the retrieval scope*
>
> *4.6 switch from exact matching to partial/containing matching to broaden the retrieval scope*

[16] In SPARQL syntax, the * is a property path operator that represents zero or more repetitions of the property path.

4.7 break down multiple-hop queries into fewer hops, to relieve the constraints of meeting all conditions across multiple hops...

3.2.7 Retrieval Recommendation Based on the Neighborhood in Ontology Subgraph

The SPARQL syntax-based recommendation approach mentioned earlier may remain insufficient. Otherwise, as discussed in Section 3.2.3, the entity extraction technique employing large language models inherently exhibits instability - the models usually extract unnecessary yet relevant entities. This characteristic paradoxically creates favorable conditions for retrieval recommendations based on ontological subgraph neighborhoods. The core methodology operates as follows: First, it identifies classes and properties within the verified SPARQL queries in Section 3.2.6. Next, it locates their positions within the ontology subgraph (see the light red part and the ticked entities within Figure 3). Finally, it expands to adjacent classes or properties from these anchor points, thereby recommending alternative SPARQL query patterns. These patterns can then be reverse-engineered to generate corresponding natural language questions.[17]

For example, consider the NLQ “东不拉分布在哪里，其所分布的地域还有哪些弹弦类乐器？”[translation: Where is the "dongbula" (dombra, 东不拉) distributed, and what other plucked string instruments are distributed in the same regions/places?] The extracted ontology subgraph is shown in Figure 3.[18] The corresponding SPARQL is as follows:

```
define input:inference 'urn:owl.ccmusicrules0214'
PREFIX bf: <http://id.loc.gov/ontologies/bibframe/>
PREFIX ctm:
<https://lib.ccmusic.edu.cn/ontologies/chinese_traditional_music#>
```

[17] This workflow corresponds to prompt8 of 2_subGraphAssemblyFromOntology_3_SPARQLgeneration.py in Git Hub

[18] (1) For the corresponding SPARQL, please refer to Instrument_Place_Instrument.sparql; (2) for the RAG result and new recommended pairs of NLQs and SPARQL , please refer to resultFromLLMsAfterPromptingWithA_question.md; (3) for the OWL code of the ontology subgraph, please refer to assembledSubgraphOfOntology_forQuestion_Instrument_Place_Instrument%26recommendation.ttl. All are in the Github repository.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?place ?placeLabel ?otherInstrument ?otherInstrumentLabel WHERE
{ # Find the dongbula instrument
  ?dongbula rdfs:label "东不拉" .
  ?dongbula a ctm:PluckedStringInstrument .
  # Find places where dongbula is distributed
  ?dongbula bf:place ?place .
  ?place rdfs:label ?placeLabel .
  # Find other plucked string instruments in the same places
  ?place ctm:placeHasMusicTypeOrInstrument ?otherInstrument .
  ?otherInstrument a ctm:PluckedStringInstrument .
  ?otherInstrument rdfs:label ?otherInstrumentLabel .
  # Exclude the dongbula itself from the results
  FILTER(?otherInstrument != ?dongbula) }
```

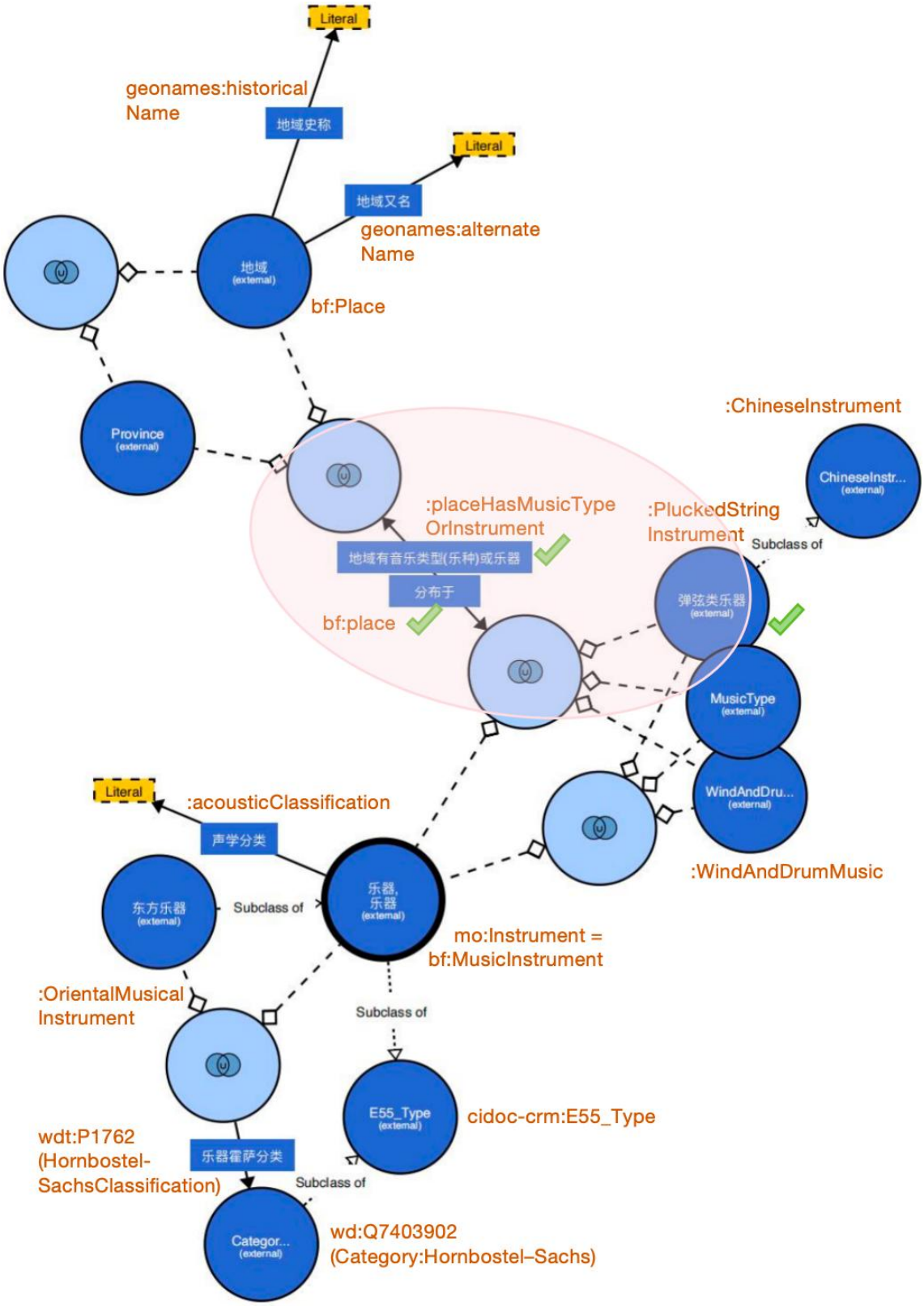

Figure 3. Visualization for Extracted Ontology Subgraph for a NLQ (using the WebVOWL 1.1.7 tool)

Blue-black nodes represent classes; Black arrows denote properties, with direction indicating class-to-property values / subject-to-object relationships; Dashed lines + square markers pointing to a light blue "double-circle intersection with embedded U" node indicate that the union of certain classes

serves as the domain or range of a property. For example, the property wdt:P1762 (Hornbostel-Sachs Classification)(乐器霍萨分类) of instruments has a domain comprising the union of mo:Instrument or :OrientalMusicalInstrument.

Reference the pink-highlighted area: This SPARQL query as above utilizes only the following classes and properties: ctm:PluckedStringInstrument, ctm:placeHasMusicTypeOrInstrument, bf:place. The latter two are inverse properties. Since the domain (or range) of one of these inverse properties is the union of MusicType, WindAndDrumMusic, and Instrument and PluckedStringInstrument, these classes are consequently implicated and incorporated into this subgraph. Hence, the extracted ontology subgraph involves broader neighboring elements, as shown in the figure above. For example: Parent classes of PluckedStringInstrument - OrientalMusicalInstrument, Instrument, cidoc-crm:E55_Type and object property - wdt:P1762 (Hornbostel-Sachs Classification), data property - acousticClassification, and place-related classes (including Province) and their data properties - geonames:historicalName, geonames:alternateName. The prompted LLMs can comprehend these relationships and thus sequentially generates a series of recommended questions (for corresponding SPARQL queries, refer to the generation log mentioned in the earlier footnote) along with answers. Here due to the limitation of space, we only display the new NLQs:

(1) What acoustic classification does Dongbula belong to, and what other instruments are in the same acoustic classification?

(2) In the areas where Dongbula is distributed, besides plucked string instruments, what other types of instruments are there?

(3) What music types or genres exist in the regions where Dongbula is distributed?

(4) What are the historical names or alternative names of the regions where Dongbula is distributed?

(5) What Chinese traditional instruments exist in the regions where Dongbula is distributed?

(6) What is the specific acoustic classification code for Dongbula?

(7) What wind and drum music exists in the regions where Dongbula is distributed?

3.3 Summary of Cases Testing

In order to evaluate its robustness, we divide the python script-based work flow into 2 parts: (1) Entity extraction from NLQ mapping with ontology, (2) Subgraph assembly and SPARQL generation, and RAG.[19] The crucial part actually lies in (1): not missing any related entity (classes and properties), even if some unnecessary entities are extracted from the OWL files. To date, through iterative debugging based on a series of question-answer cases, this ontology-subgraph-based NLQ2SPARQL workflow has achieved satisfactory results. The core testing methodology follows 3 progressive stages, as informed by the ontological class-property network (illustrated in Figure 1): First, **Binary Relation Sampling (Dyad-Level)**: Analogous to dyadic (X-Y) relationship extraction in complex networks, we systematically traverse and sample subject-predicate-object triplets (e.g., [Music Genre]-[distributed in]->[Region]), then formulate NLQs reflecting these semantic structures.

Binary Relation Sampling (Dyad-Level). Second, **Multi-Hop Chaining (Triad/K-Hop Expansion)**: Building upon last step, we compose chained questions by concatenating triplets, incrementally increasing complexity (e.g., X-Y-Z chains). Finally, **Unconstrained Questioning (Schema-Agnostic NLQs)**: Generate questions unbounded by ontological constraints, relying only on domain intuition. All question-SPARQL pairs are archived in the GitHub folder storageOfPairsOfNLQ&correspondingCorrectSPARQL&others, where filenames explicitly encode the underlying triplet structures. Three illustrative examples: The first is like Example (6) in Section 2.2. The second is in the file PieceWithPerformance_SpecialIndependentResource_Place.sparql: “哪些曲目所被收录的特藏独立资源的采录地址是杭州市？” (translation: Which music pieces were collected in SpecialIndependentResources which were recorded in Hangzhou?” The third is in the file random.sparql: “有哪些戏曲音乐在华中一带的城市分布，进而是否有曲目、曲牌和它们相关，此外，我馆是否有相应资源收藏？”[20] (Which opera music types are distributed in Central China’s cities? Are there related music pieces, *qupai* (melodic patterns), or related archival holdings in our library repository?) These

[19] See the corresponding Python code (1)“1_entitiesExtractionFromNLQ_basedOnOntology”, (2)“2_subGraphAssemblyFromOntology_3_SPARQLgeneration”. Note: the output of executing (1) consists of two lists--extracted classes and extracted properties. Pass these results to (2) as parameters, where there are tips “--corresponding to Transformed” ClassList/PropertyList.

20 Here, the LLM autonomously infers "Central China" as a geographic aggregation of provincial-level units

cases respectively align with the 3 testing stages above. By exhaustively sampling triplets with traversal, the NLQ dataset approximates a cluster sample, ensuring representativeness. During testing, the workflow were refined and any code issues triggered by specific NLQs were addressed until robust performance was achieved.

3.4 Feasibility Assessment and Improvement Direction

Finally, Evaluate the feasibility, advantages, disadvantages, and potential improvements of the method. First, the fundamental premise of the method is that the knowledge base must have a clear ontology, at least "nominally"—specifically, each instance must belong to a class (rdf:type's value). Then, a fundamental challenge in NLQ2SPARQL lies in the inherent ambiguity of natural language, which manifests in the mapping of NLQ to the ontology. For example, in a chained question, whether the premises of different sub-questions are connected by "OR" or "AND" is often implicit in natural language phrasing.[21] Another example is that entities in NLQ could be classes or instances. Such ambiguity presents both opportunities—such as revealing different semantic interpretations or facets of a question through recommendation systems—and challenges, namely introducing uncertainty in entity extraction and generating SPARQL queries that align with user intent. Therefore, unsatisfactory SPARQL retrieval due to potential ambiguity cannot always be blamed on the LLMs. Consequently, "users who know how to ask questions" often play another crucial role. In other words, the feasibility of this intelligent Q&A system also depends, to some extent, on the user's understanding of the ontology. This is one reason why this project emphasizes the role of ontology. Now, let's examine the specific advantages and disadvantages:

3.4.1 Advantages and Features & Disadvantages and Challenges

This approach converts the uncertainty inherent in entity extraction in Section 3.2.3 into the explainability and creativity of the recommendation system based on Neighborhood in Ontology Subgraph, as described in Section 3.2.7. Hence, this ontology-subgraph-

[21] For instance, in Section 3.2.7, the question "Where is the dombra distributed, and what other plucked string instruments exist in its distribution areas?" is processed by the LLMs-generated SPARQL as an "AND" relationship—it defaults to returning only regions with plucked string instruments. If the question were rephrased as "Where is the dombra distributed, and what other plucked string instruments might exist in its distribution areas?", the SPARQL might relax the query conditions using OPTIONAL or UNION which indicates "OR" relationship

driven approach can always generate results, which is valuable from a user experience perspective—avoiding the frustration of zero feedback (whether through exact matches or related recommendations) while maintaining transparency throughout the retrieval process. This transparency has a positive effect: it encourages users to gradually understand metadata ontologies and SPARQL syntax. Additionally, experiments revealed that, beyond rdfs:domain and rdfs:range, other semantics in the OWL ontology-subgraph also serve as references for LLMs-generated SPARQL and recommendations. In other words, LLMs can leverage ontological semantics to potentially incorporate ontology-based inference for data completion in SPARQL queries. To some extent, this can complement the built-in ontology-based inference mechanism.

However, when the ontology scales to a certain size (e.g., like large knowledge bases like DBpedia), the method may face challenges. In such cases, LLMs might struggle to distinguish similar properties and classes, and the workload for ontology editing and maintenance will increase simultaneously. Generating SPARQL through traversing ontology structure could be a solution.

3.4.2 Directions for Improvement

（1）On the ontology side, as LLMs' token "digestion" capacity is improving, the previous simplified version of ontology could be enriched by restoring omitted semantic information. Further optimizations include developing more automated programming solutions for ontology editing tasks as described in Section 3.2.1.

（2）On another front, entity extraction could be enhanced by limiting property scope: first filtering properties used by extracted classes, then identifying those relevant to the NLQ. The latter would leverage the technology of shapes created byVOID generator.[22]

（3）Most importantly, the current Python-coded workflow should be modularized toward an agent-based design:

A. This could also address the issue mentioned in (2) from another angle—iterative entity extraction: If the initially extracted entities are insufficient for SPARQL

[22] As noted in Section 3.1.1's literature review, shapes are schemas in the true sense, based on the closed-world assumption. For operational details, refer to https://github.com/candlecao/void-generator.

generation, the system could search for the most relevant classes or properties among the remaining entities and repeat the process until the desired achievement.
B. For the current workflow, handle potential issues like SPARQL query errors, empty returns, or timeouts during large-data queries in the SPARQL generation and RAG phases.
C. User interaction: At the retrieval stage, allow users to provide feedback on results and explanations, prompting LLMs to make necessary corrections and optimizations. Moreover, since this is a linked-data application, retrieval results may include accessible URIs. The LLMs could be prompted to visit these links during RAG to gather richer knowledge for summarization.Finally, after this phase of research, the team will conduct an in-depth review of other methods' strengths and their potential to complement this approach, building on the existing literature review.

4 Reflection

East and Southeast Asian Traditional Music cultural heritage is underrepresented, while the ontology-engaged knowledge base makes its semantics richer and more precise, thereby facilitating LLMs in the conversion of NLQ2SPARQL for more friendly retrieval. As discussed in the literature review, there are two orientations in addressing NLQ2SPARQL. Our ontology-subgraph-driven approach appears to fall under the semantic-oriented side. However, foreseeably, in the following two instances, it inevitably involves algorithms and computing strength: First, when extracting shapes from an oversized RDF database without an ontology (which can then be converted to a simulated nominal ontology)[23], considerable computing strength is still required. Second, for a retrieved ontology subgraph, graph traversal may still be necessary to test different SPARQL query patterns. In conclusion, algorithm-oriented and semantic-oriented approaches are not fundamentally distinct. Eventually, these two orientations, along with pairs of NLQ&SPARQL query examples as training shots may be integrated.

Finally, the ongoing and rapid advancement of LLMs may enable them to digest even larger ontology graphs in one go. Until then, this method essentially relies on and demands exceptionally clear data at the ontology layer, which still requires significant

[23] For how to extract the nominal ontology automatically, please also refer to https://github.com/candlecao/void-generator.

expertise and effort from information management professionals in meticulous cataloging and classification tasks. In the context of AI-driven work, we observe the interplay and evolving influence of two orientations: knowledge graphs and LLMs.

5 Acknowledgments

We acknowledge the support of Ichiro Fujinaga, Director of the Distributed Digital Music Archive and Library Laboratory in the Music Technology Area at Schulich School of Music, McGill University, as well as the LinkedMusic project team. We also appreciate the support of the Library of the China Conservatory of Music for providing data and facilitating the expansion of the Chinese Traditional Music Culture Resource Database. I also thank Jenn Riley, Associate Dean of McGill University's library, specially for her selfless help with providing abundant knowledge of the current situation of linked data application. Additionally, the early stage of this research is supported by an open fund from the Cultural Heritage Intelligence Computing Laboratory at Wuhan University, a China Philosophical and Social Science Laboratory (Project No. 2023ICLCH010). The author also thank Tao Chen, Associate Professor and Zhenyu Hu, a student reading for her master's degree, at the School of Information Management, Sun Yat-sen University, for their valuable assistance.

6 References

[1] Caio Viktor S. Avila, Vania M.P.Vidal, Wellington Franco, Marco A. Casanova. "Experiments with text-to-SPARQL based on ChatGPT". 2024 IEEE 18th International Conference on Semantic Computing (ICSC), 2024.

[2] Davide Varagnolo, Dora Melo, Irene Pimenta Rodrigues. "An Ontology-Based Question_Answering, from Natural Language to SPARQL Query". KEOD 2023 - 15th International Conference on Knowledge Engineering and Ontology Development, 2023.

[3] Dean Allemang, Juan F. Sequeda. "Increasing The LLM Accuracy for Question Answering Ontologies to the Rescue". arXiv, 2024.

[4] Dmitrii Pliukhin, Daniil Radyush, Liubov Kovriguina, Dmitry Mouromtsev. "Improving Subgraph Extraction Algorithms for One-Shot SPARQL Query Generation with Large Language Models". ISWC 2023: Scholarly QALD Challenge, November 6-10, 2023, Athens, Greece.

[5] Jingfang Yuan. "Chinese Music Species". People's Publishing House, 2023.

[6] Junjun Cao, Junxuan Li, Yuhe Wang. “Research on the Construction of ‘Metadata Ontology’ for Chinese Traditional Music Knowledge Base”. Digital Humanities Research, 2022.
[7] Junjun Cao, “Research on the Construction of Traditional Chinese Music Culture Knowledge Base from the Perspective of ‘Digital Humanities’” (Postdoctoral Research Report). China Music Conservatory, 2024.
[8] Liubov Kovriguina, Roman Teucher, Daniil Radyush, Dmitry Mouromtsev. “SPARQLGEN: One-Shot Prompt-based Approach for SPARQL Query Generation”. SEMANTICS 2024 EU: 19th International Conference on Semantic Systems, September 20-22, 2023, Leipzig, Germany.
[9] Shuangtao Yang, Mao Teng, Xiaozheng Dong, Fu Bo. “LLM-Based SPARQL Generation with Selected Schema from Large Scale Knowledge Base”. Extracted from Haofen Wang, Xianpei Han, Ming Liu, Gong Cheng, Yongbin Liu, Ningyu Zhang (Eds). Knowledge Graph and Semantic Computing (Knowledge Graph Empowers Artificial General Intelligence). Springer, 2023.
[10] Vincent Emonet, Jerven Bolleman, Severine Duvaud, Tarcisio Mendes de Farias, Ana Claudia Sima. “LLM-based SPARQL Query Generation from Natural Language over Federated Knowledge Graphs”. ISWC’24, Special Session on LLMs, Slot 2: Bridging the Gap: Building Knowledge-Enhanced Gen AI, 11-15 Nov 2024.
[11] Yaohua Wang, Yaxiong Du, Zhou Wang, Fulin Liu. “Introduction to Traditional Chinese Music”. People’s Music Publishing House, 2022.
[12] Yaxiong Du. “Overview for Folk Music of Minority Ethnic Groups in China”. Shanghai Music Conservatory Publishing House, 2014.
[13] Youqin Ying, Keren Sun. “Chinese Music Grand Dictionary”. Education Publishing House of Shanghai Century Publishing Corporation, 2015.