Ontology-based Multimedia Contents Retrieval Framework in Smart TV Environment

Moohun Lee*, Joonmyun Cho*, Jeongju Yoo*, Jinwoo Hong*

*Next Generation SmartTV Research Department,

ETRI (Electronics and Telecommunications Research Institute), Korea

leemh@etri.re.kr, jmcho@etri.re.kr, jjyoo@etri.re.kr, jwhong@etri.re.kr

***Abstract*—Semantic search promises to provide more accurate result than present-day keyword matching-based search by using the knowledge represented logically (i.e., knowledge base). But, the ordinary users don't know well the complex formal query language and schema of the knowledge base. So, the system should interpret the meaning of user's keywords. Such requirements are conspicuous especially in smart media such as smart phone, IPTV and smart TV. In this paper, we describe a framework for the semantic retrieval of multimedia contents. Our framework is ontological knowledge base-driven in the sense that the interpretation process is integrated into a unified structure around a knowledge base, which is built on domain ontologies. Our framework also integrates such components as for knowledge base augmentation by user preference and context. We also present our prototype system we have been developing to test our framework ideas.**

***Keyword*—Knowledge Base, Ontology, Semantic Search, Smart TV**

I. Introduction

WE are witnessing the last few years rapid increasing of smart media such as smart phone, tablet PC and smart TV. Especially, smart TV equipped with a processor, large inexpensive disk storage, and high-speed Internet access is predicted to be an open home digital hub in the near future [1]. In the line of such movement, digital audiovisual content libraries and video on demand services are already making business in the market. Furthermore, the business will be expanded to the broadcasting contents by real-time program capturing and streaming capabilities [2].

Recently, the advanced technology of contents retrieval becomes a dominant factor for the success of such business [1,3,4]. For example, users want to easily search for contents regardless of the sources (i.e. repositories) such as digital contents libraries, VoD(Video on Demand) services, and broadcasting. Moreover, users expect that their preference and context such as favorite football player, favorite football team, the present football competition and major games of the competition are reflected in their search results spontaneously.

The above requirements can be satisfied in a system which stores semantic descriptions for the multimedia contents, user preferences, and contexts and provides semantic search capabilities on the semantic descriptions [1,3-8]. However, considering the requirements, we can see some features that the system should have. First, since there are many different content sources and metadata representation schemes, the semantic descriptions for the contents should be managed in a single repository (i.e. knowledge base) to support integrated semantic search. Second, the users should be able to express their information need as a simple keyword phrases, because ordinary users don't know the complex formal query language and the schema of the underlying knowledge base. Third, the knowledge of user preference and context having very dynamic trait should be represented in the same way with the knowledge of contents metadata and treated equally in order to reflect them automatically.

The input of the semantic search system is a keyword phrase and the final output is a list of contents matching with the keyword phrase reflecting user's intention. In this process the system has to interpret the keyword query to fill the gap between the keywords and the user's intention and has to find the metadata description in the knowledge base that accord with the interpretation [7-10]. The target contents are those connected to the identified metadata description. In other words, the process converts the keyword phrase into a formal query statement of the knowledge base. Due to the ambiguity of keyword phrase, there may be multiple formal queries produced from one keyword query. How to rank these queries is a big challenge[7].

In this paper, we describe a framework for semantic multimedia contents retrieval in smart TV. Our framework is

Manuscript received June 29, 2012. This work was supported by the ETRI R&D Program of KCC(Korea Communications Commission), Korea [11921-03001, "Development of Beyond Smart TV Technology"].

Moohun Lee is with the Next Generation SmartTV Research Department, ETRI, 218 Gajeongno, Yuseong-gu, Daejeon, Rep. of Korea (corresponding author to provide phone: 82-42-860-1826; fax: 82-42-860-6202; e-mail: leemh@etri.re.kr).

Joonmyun Cho is with the Next Generation SmartTV Research Department, ETRI, 218 Gajeongno, Yuseong-gu, Daejeon, Rep. of Korea (e-mail: jmcho@etri.re.kr).

Jeongju Yoo is with the Next Generation SmartTV Research Department, ETRI, 218 Gajeongno, Yuseong-gu, Daejeon, Rep. of Korea (e-mail: jjyoo@etri.re.kr).

Jinwoo Hong is with the Next Generation SmartTV Research Department, ETRI, 218 Gajeongno, Yuseong-gu, Daejeon, Rep. of Korea (e-mail: jwhong@etri.re.kr).

ontological knowledge base-driven in the sense that the interpretation process is integrated into a unified structure around a knowledge base built on domain ontologies. Moreover, our framework allows the knowledge base augmentation with the dynamic context and user preference knowledge in a uniform manner. The main objectives of our framework are to provide enhanced retrieval performance and better user interfaces based on the ontological knowledge base. Our framework provides domain ontologies to represent logically the contents metadata, user preference, and context data. It also includes such components as for knowledge base augmentation, user query interpretation, and adjustment of the interpretation result.

This paper is structured as follows: In Section 2 we briefly explain the conceptual architecture and principal functional components of our framework as well as generic approaches for the functions. Then, the prototype system to verify our framework ideas is presented in Section 3. In Section 4 we describe an application of prototype system in smart TV environment. Section 5 outlines the related works. In Section 6 we provide the conclusions of our work and a brief description of our future works.

II. Framework Description

In this section we provide an outline of our framework for the semantic retrieval of multimedia contents in smart TV. The conceptual architecture of the framework is presented in Figure 1.

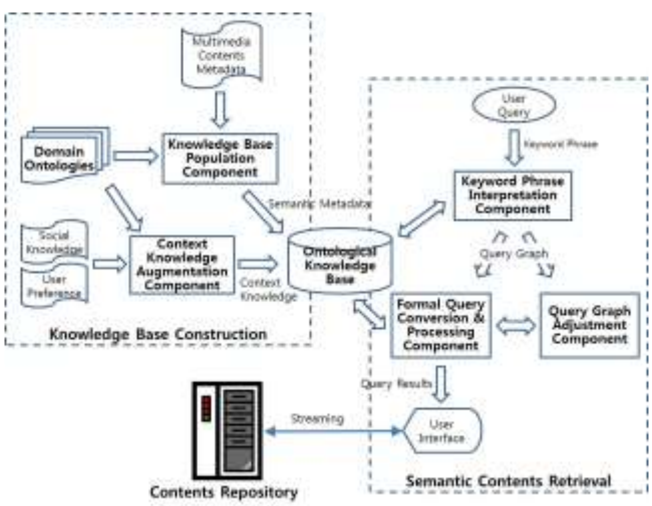


Fig. 1. Conceptual architecture of the framework

The framework has two sections. The first section relates to knowledge base construction, and the second one relates to semantic contents retrieval. The major functional components of the first section are domain ontologies, a knowledge base population component and a knowledge augmentation component. The second section has, as its major functional components, a keyword phrase interpretation component, a formal query conversion & processing component, and a query graph adjustment component.

As described in the figure, our framework is ontological knowledge base-driven as all the components provide their functions based on the knowledge base which is built on the domain ontologies. More specifically, the metadata of contents is represented in a logical graph in the knowledge base using the domain ontologies of such as movie, drama, and sports. The user preference and context data are also represented in the same way as the metadata of contents, and integrated in the knowledge graph. The user query (i.e. keyword phrase) is interpreted into a query graph by the keyword phrase interpretation component, and then a list of contents that accord with user's intention is returned by the formal query conversion & processing component. The keyword phrase interpretation is performed by picking up the most likely partial knowledge graph (called query graph) among candidate partial graphs that are extracted from the knowledge base. The candidate partial graphs are those that the user query could denote. Since the interpretation result is given as a graph, the user can easily adjust it to be fit for his/her intention in GUI.

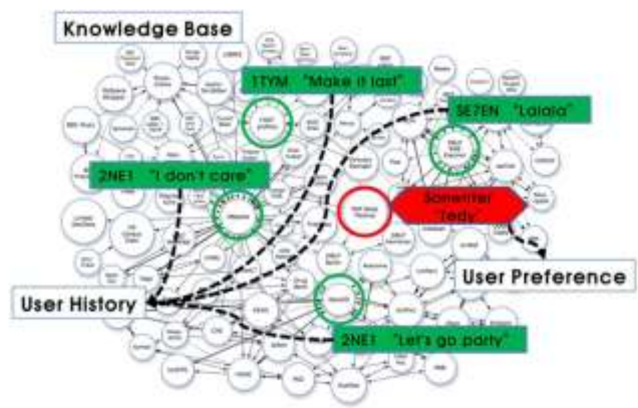


Fig. 2. User preference learning based on KB

Our approach for the knowledge augmentation with user preference is as follows. We can acquire the user preference knowledge through two ways. The first one is to get it from the user profile that user specified explicitly. The second one is to learn it from user's query history. Our approach for the user preference learning is distinct from general association rule mining, because our approach is also based on the knowledge base. In usual association rule mining, system mines only the gathered data for association rules. In our approach, the system mines whole knowledge base. That means the association rules are acquired not from the gathered query history but from the knowledge having been expanded through the knowledge graph. For example, consider the case that the history data about music video query which consists of singer (or group) name and music title is gathered (e.g., "2NE1, I don't care", "SE7EN, Lalala", "1TYM, Make it last", "2NE1, Let's go party"). We can map each element of the gathered data to knowledge entity so that we can expand the history data to the knowledge such as composer, genre and member of the music group. In the end we can find that the user's preference is for the composer "Tedy". Figure 2 shows the concept of our approach.

On the other hand, the social knowledge is acquired by applying text mining techniques to social media such as blog, news and bulletin board. For example, we can acquire the knowledge that the term, "Ronaldo Wink" relates to the quarterfinal match between Portugal and Germany in the World Cup Germany 2006 by analysing blogs and news. With the knowledge base being augmented by such knowledge, the system can give much more accurate query results to the user when the user enters the keyword phrase, "Ronaldo wink football video".

The user query interpretation process in our approach, which is basically a graph-based technique like as other research [7-10], converts the keyword phrase into a partial knowledge graph of the knowledge base. A query graph can be easily translated into a SPARQL[11] statement, because SPARQL is a graph pattern based query language. The process consists of three steps: term mapping, query graph construction, query graph ranking. Term mapping maps the terms of the keyword phrase to the entities of the knowledge base. After that, query graph construction links the mapped entities so that the missing relations and concepts can be obtained and a complete query graph can be constructed. Finally, the query graph ranking estimates the most likely query graph that best accords with the user's intention among all the candidate query graphs.

III. Prototype System

In this section we present the prototype system that has been developing to verify our framework idea, focusing on main components such as domain ontology, ontological knowledge base, and user query interpretation module.

A. Movie Domain Ontology

In order to evaluate our framework, we have designed an ontology for the description of movie contents using Web Ontology Language (OWL)[12]. Figure 3 shows a part of the ontology. In the left section of the figure you can see the hierarchy of classes and object property. You can see the description of LeadingActor class in the right section. In OWL, which is based on Description Logic (DL), classes are classified into two categories: primitive class and defined class. A primitive class has a necessary membership condition, while a defined class has a necessary and sufficient membership condition [12]. In the figure, defined classes are marked with triple line icon.

In our approach, the usage between primitive and defined class is distinguished strictly. Though OWL allows the ontology modellers to assert an individual to be an instance of multiple classes, we assert an individual to be an instance of one class. They are automatically asserted to be an instance of other classes afterwards by DL reasoning. For example, every individual that denotes a human being must be asserted to be an instance of Person class, and some of them having playLeadingRole property are asserted to be instances of LeadingActor class afterwards by DL reasoning engine. Such scheme means role concepts ought to be represented differently from type concepts. A role concept is a class, but, unlike a type class, its instances are not fixed [13]. For example, the instances of LeadingActor class continuously change, as new movies are added to the knowledge base or existing movies are removed. A role concept must be related to an event its instance participates in. The properties used in the necessary and sufficient membership condition of a defined class can be regarded as the events.

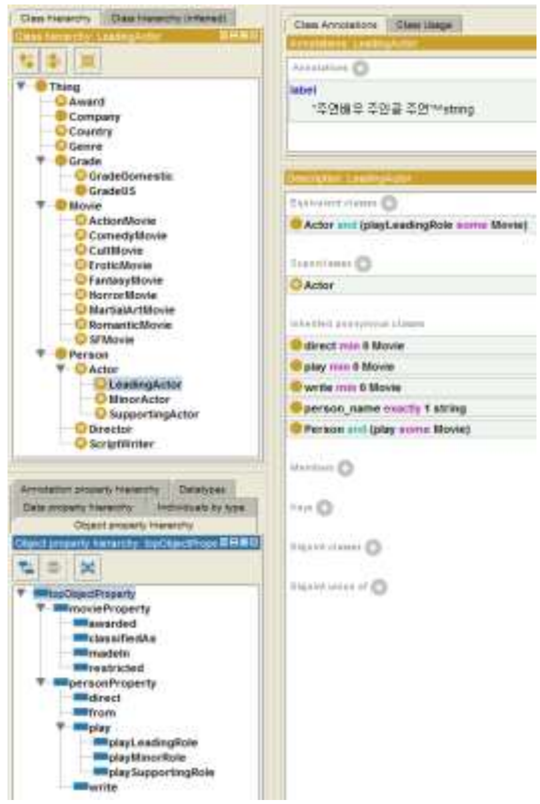


Fig. 3. Movie domain ontology

It is very difficult and troublesome to build a knowledge base from scratch. So an automated or semi-automated tool is needed to migrate contents metadata from a legacy database to the knowledge base. The above scheme makes the knowledge base population including migration very easy. For example, consider an individual denoting a movie content, Iron Man. Under the above scheme it is enough to assert the individual as an instance of Movie class and describe its properties. You don't need to suffer from deciding which other classes the individual should be belonged to. The reasoning engine automatically asserts the individual as an instance of ActionMovie and SFMovie classes by using the described properties.

B. Movie Contents Knowledge Base

Figure 4 shows the knowledge base built using the movie domain ontology. In the middle section of the figure you can see the instances of ActionMovie class, while in the right section you can see the property description of movie_0002 individual. The individuals are automatically classified as instances of ActionMovie class by the DL reasoning engine.

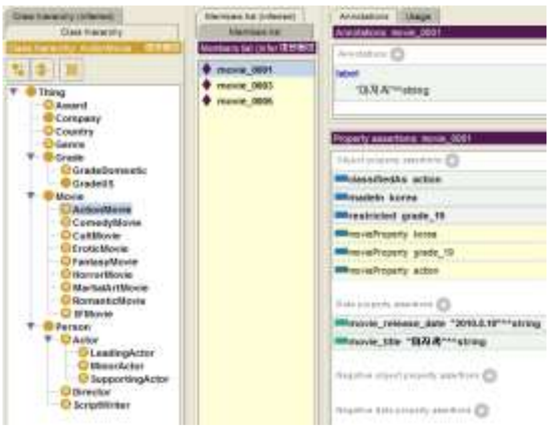


Fig. 4. Movie Contents Knowledge base

In our knowledge base, every entity has a label annotation as in upper-right section of the figure. Using such labels the system makes indexes of the entities and maps user keyword to the corresponding entities. Figure 5 shows the process for knowledge base indexing and user query interpretation.

C. User Query Interpretation

The keyword phrase interpretation component translates the user query into a partial knowledge graph (i.e. query graph). That process is carried out through three steps as in Figure 5: term mapping, query graph construction and query graph ranking.

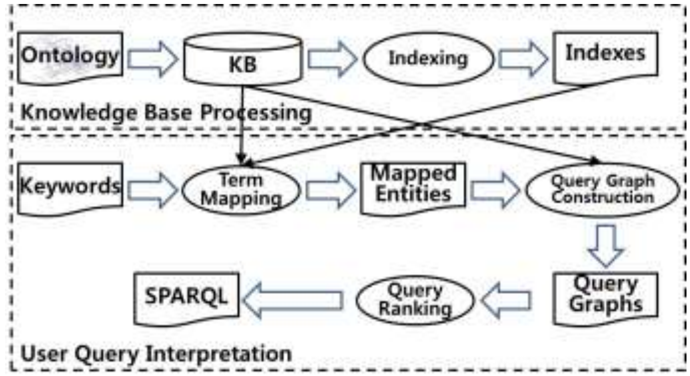


Fig. 5. KB processing & User Query Interpretation Process

Term Mapping

The purpose of term mapping is to find corresponding knowledge entities (i.e. classes, individuals, properties and literals) for each term in the keyword query. The name and labels of the entities are used for mapping. In our implementation, the term mapping process indexes the names and labels of knowledge entities, and using the fuzzy search feature of Lucene[14], returns a list of knowledge entities for each entered keyword. The returned entities are ranked according to syntactic similarity to the respective keyword.

```
USER_QUERY_INTERPRETATION(keyword_list)
   user_query_segment_mapping_list = TERM_MAPPING(keyword_list);
   initialize new empty query_graph;
   initialize new empty instance_entity_list;
   FOR mapping in user_query_segment_mapping_list
      add instance_entity_list to mapping;
      clear instance_entity_list;
      query_graph_store = CONSTRUCT_QUERY_GRAPH(mapping);
  extract shortest_query_graph containing all the knowledge_entities of mapping
        from query_graph_store;
      add shortest_query_graph to query_graph;
      IF mapping is not the last one
         THEN translate shortest_query_graph into SPARQL query statement;
               instance_entity_list = process SPARQL query with KB;
   RETURN query_graph;

TERM_MAPPING(kwyword_list)
   initialize new empty user_query_segment_mapping_list;
   FOR keyword in kwyword_list
      initialize new empty mapping_list;
      find knowledge_entity for keyword;
      add knowledge_entity to mapping_list;
      IF knowledge_entity is class entity
         THEN add mapping_list to user_query_segment_mapping_list;
   RETURN user_query_segment_mapping_list;

QUERY_GRAPH_CONSTRUCTION(mapping)
   initialize new empty query_graph_store;
   extract class_entity from mapping;
   FOR knowledge_entity (except class_entity) in mapping
      make connections between knowledge_entity and class_entity;
      add connections to query_graph_store;
   RETURN query_graph_store;
```

Fig. 6. User query interpretation algorithm

Query Graph Construction

The query graph construction process builds up candidate query graphs with the knowledge entities mapped in the term mapping step. The process explores all entities related to the mapped entities. The exploration simply incorporates all entities and literals within a pre-defined range (i.e. distance) and makes up the missing relations and concepts for the user query. From the explored paths, only those, called connections, are selected where the first and the last vertex correspond to one of the mapped entities. When the connections can be joined into a graph containing all the entities mapped in the term mapping step, the candidate query graph is generated according to the following rules: 1) Class entities mapped by the term mapping or discovered by the exploration are regarded as variable nodes. 2) Individual entities and literals are regarded as end nodes. 3) Property entities are regarded as the edges of the query graph.

Query Graph Ranking

After the term mapping and query graph construction, multiple candidate query graphs will be produced. There comes the problem: how to pick up the most appropriate query graph for the user? Various models have been suggested for the evaluation [7-10,15]. One of the simplest way is to give higher rank to the query graph that has the shorter path connecting all the mapped entities. We use the shorter path model because the assumption is generally valid that the knowledge entities which have conceptual relations are located close to each other in a knowledge graph [9].

The algorithm for the user query interpretation process is shown in Figure 6. In the algorithm we consider the case that a keyword is mapped to one entity. But actually it is not, so the multiple mapped entities should be split into different query sets. The query set split step is a process of enumerating all possible combinations from different senses of each keyword.

IV. IMPLEMENTATION OF PROTOTYPE SYSTEM

In this section we describe the implementation of prototype system on smart TV environment. There should be approximately 2,000 instances of content in ontological knowledge base. And they are an experimental metadata for the performance verification of prototype system. The user keyword phrase can be interpreted by prototype system in various ways. It is possible for smart TV environment to interpret the search intention of their keyword phrase and knowledge base. The interpreted user queries and search results of knowledge base can be used in combination with other contents searcher, such as VoD, Broadcast, and UCC(User Created Contents) searcher. The prototype system consists of three parts: the Knowledge Base Server, Content Retrieval Server, and Smart TV Client.

A. Knowledge Base Server

The knowledge base server enable contents retrieval server to perform exploration of knowledge base. The knowledge base consists of contents metadata about instances of movie, drama, entertainment, and music video.

The knowledge base server should be to create the index about entity-label. Then, it should be to recognize the entity for entered user keyword phrase. After that knowledge base server links the recognized entities so that query graph can be constructed. And the knowledge base server should be to rank candidate query graphs. At that time, it is to make use of the model of entities weight.

Knowledge Entity Indexing

The knowledge base server loads the contents metadata. The server is configured with an entity label and the URI(Uniform Resource Identifier) to generate the index. This index is used to recognize the knowledge entity to user keyword in the process of knowledge entity recognition. The creation of indexes in the implementation of the lucene library is used. The index is divided into a resource and literal.

Knowledge Entity Recognition

The purpose of Knowledge Entity Recognition is to find corresponding ontology entities, such as classes, individual, properties and literals. In our implementation, we are using Korean morphological analyzer. After the completion of stemming, find the best combination of keyword for entity recognition. Morphological Analyzer is a software library capable of detecting morphemes in a piece of keyword phrase. The Figure 7 describes the results of knowledge entity recognition from the best combination of keywords. The results of recognized entities have contained node URI and type information of node.

Fig. 7. Results of knowledge entity recognition

Query Graph Construction and Ranking

The knowledge base server builds up query combination from the recognized entities. The query combination process constructs candidate query graphs with the ontology resources mapped entities. The query graph ranking is to find the best query graph from candidate query graphs. Figure 8 shows the conceptual spanning tree of user query. Figure 9 shows an expanded ontological query graphs. It is expanded the query graph through the path of graph from the conceptual spanning tree. It is the best query graph.

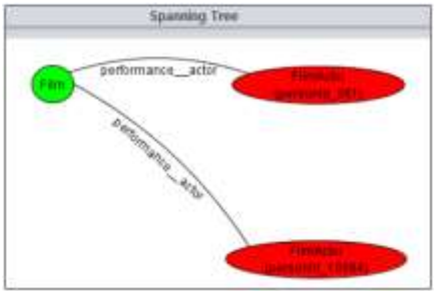


Fig. 8. Conceptual spanning tree of user query

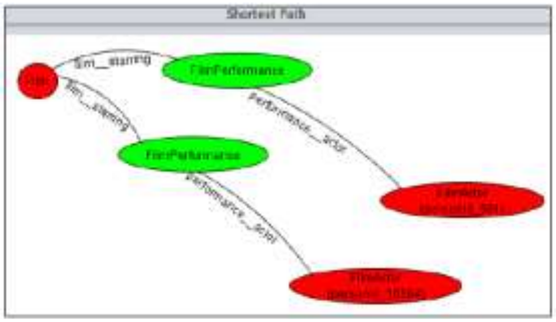
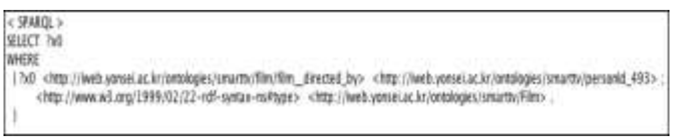


Fig. 9. Expanded ontological query graphs

Dynamic SPARQL Generation

The knowledge base server performs the dynamic SPARQL generation through the best query graph. Figure 10 shows an example of SPARQL query. In our implementation we are using Jena ARQ that is SPARQL processing engine.



```
< SPARQL >
SELECT ?x0
WHERE
{ ?x0 <http://iweb.yonsei.ac.kr/ontologies/smarttv/film/film__directed_by> <http://iweb.yonsei.ac.kr/ontologies/smarttv/personId_493> ;
    <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://iweb.yonsei.ac.kr/ontologies/smarttv/Film> .
}
```

Fig. 10. Example of SPARQL

B. *Contents Retrieval Server*

The contents retrieval server is passed the user entered keywords. And interpreted query is created by knowledge base server. The interpreted query is passed to the VoD Contents Searcher, Broadcast Contents Searcher, and UCC Contents Searcher. The search results are returned by the each searcher. Figure 11 shows the class diagram of contents retrieval server.

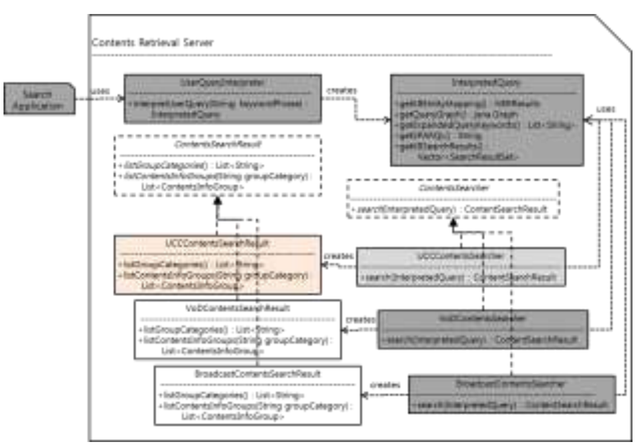


Fig. 11. Class diagram of contents retrieval server

User Query Interpreter

The user query interpreter creates an interpreted query via knowledge entity recognition, query graph construction and ranking, dynamic SPARQL generation and SPARQL processing results.

VoD Contents Searcher

The VoD Contents Searcher retrieves contents of VoD to provide streaming services in VoD contents repository. The search results are applied to the dynamic classification system.

Broadcast Contents Searcher

The Broadcast Contents Searcher retrieves contents of broadcast in EPG(Electronic Program Guide) database.

UCC Searcher

The UCC Searcher generates a query that is suitable for global repository, such as Youtube and Daum TV pot. The created query is a global repository input. And search results are completed post-processing, such as filtering, ranking, and grouping.

C. *Smart TV Client*

The smart TV client acquires the user entered keywords, and it is passed on to the contents retrieval server. The user query interpreter delivers the interpreted query for user feedback. Also, it provides the search results via filtering, ranking, and grouping.

Smart TV Search Interface

Figure 12 describes the GUI of smart TV client. The smart TV client consists of three parts: the preferred broadcast channels, information of preferred broadcast contents, and history of VoD contents. User entered keywords is passed to the contents retrieval server.

Fig. 12. Search interface in smart TV

Smart TV Search Results Interface

The search results interface provides two type of information. Firstly, the client provides search results to be filtered, ranked, and grouped. Second, it provides expanded keywords phrase for interpretation of user keyword. Figure 13 shows search results interface in smart TV.

Fig. 13. Search results interface in smart TV

Smart TV Recommendation Interface

Figure 14 shows recommendation interface for preferred broadcast contents. In order to provide more accurate program recommendation, we use not only the user watching history, but also the user program preference and program preference updated as a user preference profile by periods.

Fig. 14. Personalized video recommendation interface in smart TV

V. Related Works

Recently, digital audiovisual library systems, video on demand service systems are developed based on widely accepted standards for audiovisual content descriptions such as MPEG-7 [16] and TV-Anytime [2]. An MPEG-7 driven framework for managing semantic metadata for audiovisual content was presented in [1]. The framework allows the description of domain ontologies within MPEG-7 and the encoding of semantic multimedia metadata descriptions that utilize the domain ontologies in the Semantic DS of the MPEG-7 MDS.

On the other hand, translating keywords to formal queries is a line of research of information retrieval communities. [7] represents an attempt that specifically deals with keyword queries in semantic search engines. There, keywords are mapped to elements of triple patterns of predefined query templates. These templates fix the structure of the resulting queries a priori. These problems have been tackled recently by [8.9]. In [9], a more generic graph-based approach has been proposed to explore all possible connections between nodes that correspond to keywords in the query. This way, all interpretations that can be derived from the underlying knowledge graph can be computed.

With respect to these recent works, our approach is distinct in three aspects. First, we enrich the knowledge graph with user preference knowledge and context knowledge. Thus, the meanings of user query are interpreted more accurately. Second, our framework provides the principle for ontology design and knowledge base population so that contents' metadata can be easily migrated by the computer from legacy content libraries. Third, user can easily adjust the interpretation result of his/her keyword query in GUI, since the interpretation result is provided as a partial knowledge graph.

In Semantic Web, the semantic information of the Web is recorded by RDF triple and is embedded in Web pages. In RDF triple, the concepts and their relationships are defined. We call the data defining the resource and its relations (concept and property) metadata. Relations between concepts/instance are required to be explicitly stated in formal logic queries, which are often missing in keyword queries. OntoLook is a prototype relation-based search engine, which has been implemented in a virtual Semantic Web environment. The core idea of "OntoLook" is that there are relations among the submitted keywords, and Semantic Web offers the ability of processing relations at the system architecture level[17].

VI. Conclusion and Future Works

In this paper, we proposed an ontological knowledge base-driven framework for semantic multimedia contents retrieval. We briefly explained the conceptual architecture and principal components. We also explained generic approaches to implement each component. Our framework has the distinctive features that the knowledge of user preference and context is spontaneously reflected in the query process and that the interpretation of user query can be easily adjusted by the user in GUI. Such features stem from the fact that all the knowledge of content metadata, user preference and context is seamlessly represented in a single knowledge graph and that the interpretation result is also represented as a partial knowledge graph.

To verify our framework ideas we have been developing a prototype system. We discussed the domain ontology, i.e. movie ontology and some principles for ontology design and knowledge base construction. The principle is simple but very powerful when we can use Description Logic reasoning. We gave an illustration of the knowledge base which is built based on the movie ontology and showed an example of DL reasoning in use. The interpretation process of user query (i.e. keyword phrase) was explained and the algorithm for the process was provided.

Our prototype is not at the final version, but still developing. In the future we will implement the knowledge augmentation component and the query graph adjustment component. Especially, the user preference learning from user query history and the social knowledge mining would give distinctive features to our framework.

REFERENCES

[1] C. Tsinaraki, E. Fatourou and S. Christodoulakis, "An Ontology-Driven Framework for the Management of Semantic Metadata describing Audiovisual Information", in Proceedings of the 15th International Conference on Advanced Information Systems Engineering (CAiSE'03), 2003.

[2] TV-Anytime Forum, http://www.tv-anytime.org

[3] E. Mäkelä, "Survey of Semantic Search Research", in Proceedings of the Seminar on Knowledge Management on the Semantic Web, 2005

[4] H. Wang, K. Zhang, Q. Liu, T. Tran and Y. Yu, "Q2Semantic: A Lightweight Keyword Interface to Semantic Search", in Proceedings of the 5th European semantic web conference on The semantic web: research and applications (ESWC'08), 2008

[5] E. Hyvönen, S. Saarela and K. Viljanen, "Ontogator: Combining View- and Ontology-Based Search with Semantic Browsing", in Proceedings of XML Finland 2003, 2003

[6] M. Lux, W. Klieber and M. Granitzer, "Caliph & Emir: Semantics in Multimeida Retrieval and Annotation", in Proceedings of the 19th International CODATA Conference, 2004

[7] Y. Lei, V. Uren and E. Motta, "SemSearch: A Search Engine for the Semantic Web", LNCS(LNAI) vol.4248, 2006

[8] Q. Zhou, C. Wang, M. Xiong, H. Wang and Y. Yu, "SPARK: Adapting Keyword Query to Semantic Search", LNCS vol. 4825, 2007

[9] T. Tran, P. Cimiano, S. Rudolph and R. Studer, "Ontology-Based Interpretation of Keywords for Semantic Search", LNCS vol.4825, 2007

[10] J. Royo, E. Mena, J. Bernard and A. Illarramendi, "Searching the Web: from keywords to semantic queries", in Proceedings of the Third International Conference on Information Technology and Applications (ICITA 2005), 2005

[11] SPARQL Query Language for RDF, W3C, http://www.w3.org/TR/rdf-sparql-query/

[12] OWL Web Ontology Language, W3C, http://www.w3c.org/TR/owl-ref/

[13] N. Guarino, C. Welty, "A formal ontology of properties", in Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management, LNCS Lecture Notes in Computer Science (LNCS) vol. 1937, 2000

[14] Lucene, Apache Software Foundation, http://lucene.apache.org/java/docs/index.html

[15] N. Stojanovic, R. Studer and L. Stojanovic, "An Approach for Step-By-Step Query Refinement in the Ontology-Based Information Retrieval", in Proceedings of the IEEE/ACM Web Intelligence Conference (WI), New York, 2004

[16] MPEG-7 (Multimedia Content Description Interface), MPEG Group, http://mpeg.telecomitalialab.com/standards/mpeg-7/mpeg-7.htm

[17] Y. Li, Y. Wang, X. Huang, "A Relation-Based Search Engine in Semantic Web", Proceedings of IEEE Transactions on Knowledge and Data Engineering 19(2), 2007

Moohun Lee received his BS and MS degree in computer engineering from Hannam University, Daejeon, Korea, in 2002 and 2004, respectively. He joined ETRI, Daejeon, Rep. of Korea in 2008 and was involved with the OPRoS (Open Platform for Robot Service) project until 2010. He is currently working as a senior researcher in smart search and recommendation system development for smart TV project. His research interests include ontology-based knowledge base system, semantic search, and information retrieval.

Joonmyun Cho received his BS, MS and PhD in mechanical Engineering from Korea Advanced Institute of Science and Technology (KAIST) in 1993, 1995 and 2006, respectively. He joined ETRI, Daejeon, Rep. of Korea in 2007 and was involved with the URC (Ubiquitous Robotic Companion) project until 2011. He is currently working as a project leader in smart search and recommendation system development for smart TV project. His research interests include ontology-based knowledge base system and intelligent agent system.

Jeongju Yoo received the BS and MS degree in Telecommunications in 1982 and 1984, respectively, from Kwangwoon University, Seoul, Korea. He received the Ph.D degree in Computing Science from Lancaster University, United Kingdom in 2001. Since 1984, he has been a Principal Member of Technical Staff in the Next Generation Smart TV Research Department of Electronics and Telecommunications Research Institute (ETRI), Korea. He was a Head of MPEG Korea delgates from 2007 to 2009 and he is a Director of Smart TV Media Research Team at ETRI. His research interests are in the area of QoS, video coding, media streaming, and multiscreen service technology of Smart TV.

Jinwoo Hong received the BS and MS degrees in electronic engineering from Kwangwoon University, Seoul, Korea, in 1982 and 1984, respectively. He also received the PhD in computer engineering from the same university in 1993. Since 1984, he has been with ETRI, Daejeon, Korea, as a principal member of engineering staff, where he is currently a director of the Next Generation SmartTV Research Department. From 1998 to 1999, he conducted research at Fraunhofer Institute in Erlangen, Germany, as a visiting researcher. His research interests include multimedia framework technology, broadcasting media and service, personal broadcasting, and realistic media.