Structuring and Querying Personalized Audio Using Ontologies

Latifur Khan
Department of Computer Science and Integrated Media Systems Center
University of Southern California
Los Angeles, California 90089
latifurk@usc.edu

1. INTRODUCTION

The huge amount of information available via electronic means can easily overwhelm end-users. Further, the transfer of any irrelevant information over the network to end-users wastes network bandwidth. Therefore, the need for user-customized information selection is clear. The goal in customized selection and delivery is high precision (little irrelevant information) and high recall (not missing relevant information).

We propose to design, implement, and experimentally test a user-customized audio information-on-demand system, which we term Personal AudioCast (PAC). Our vision here is a facility to select and deliver customized information from a dynamic database, in audio form. PAC would introduce a new level of empowerment for news and information consumers;. beyond simply allowing users to select stories according to topics or keywords. Personal AudioCast would build custom 'broadcasts' tailored to a user's interests, communication style, past choices, and even their physical location.

To facilitate access to a particular segment of audio or video, two main approaches have been employed: fully automated content indexing and manual annotation. Due to the weakness of the content analysis algorithm, especially in audio/speech, we have chosen the latter approach. We need to explicitly capture the description of audio to allow appropriate selection and presentation. We term these descriptions **metadata**. Broadcast audio consists of multiple news items. Some news items are of interest to the user and some are not. Therefore, we need to index the long audio so that we can access a particular meaningful segment(s). To capture the structure of audio and to retrieve segments, we propose a model that supports the notion of arbitrary attributes. Furthermore, for the query language, we rely entirely on conventional SQL; however, we do not propose a new language from scratch.

For a greater exploitation of metadata, it is necessary to support their correlation. This support can be provided by ontology, that is, a specification of a conceptualization. Ontology defines a set of representational terms (concepts) and the relationships among them to describe a target world. In this paper, we construct a domain dependent ontology. Our ontology facilitates: metadata acquisition of audio segment, expression of user interests, generation of information selection requests and novel optimization techniques that improve query performance. Ontology provides an abstract and flexible way to query the system which cannot be achieved by simple keyword search. In other words, it provides richer forms of information upon which users can base their query.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advant -age and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ACM Multimedia '99 (Part 2) 10/99 Orlando, FL, USA
© 1999 ACM 1-58113-239-5/99/0010...$5.00

2. PROPOSED SOLUTION

Since audio is by nature serial, random access to audio information may be of limited use. To facilitate access of useful segments of audio information, we need to identify entry points/jump locations in the audio recording. A change of speaker and long pauses can serve to identify entry points. By employing the above strategy, we can detect the boundaries of audio segments. An audio object is composed of a sequence of contiguous segments. In our model, the audio object's interval is defined by the starting time and ending time. Further, an audio object is described by a set of self-explanatory tags along with values.

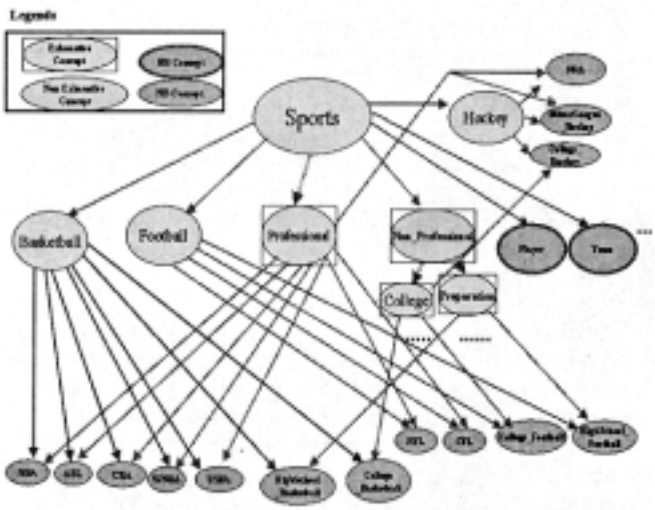

Figure 1. A Small Portion of Ontology for Sports Domain

To facilitate annotation of audio objects, tags and values come from a domain dependent ontology (sports). The ontology (Fig. 1) is described by a directed acyclic graph (DAG). Each node in the

DAG represents a concept. Arcs represent association between concepts. For example, "NBA" is a sub-concept of "Basketball," the association between them is IS-A. In general, each concept contains tag, value(s) and a synonymous list. Note that this tag or label name is unique in the ontology. In addition, the synonymous list of a concept contains vocabulary for which the concept is matched with user requests. Each interior node in the DAG is divided into two categories: exhaustive (EX) and non-exhaustive (NEX). If a concept is entirely partitioned into a set of sub-concepts, then we call this concept **exhaustive (EX)**. On the contrary, when a concept cannot entirely be partitioned among sub-concepts, it is termed **non-exhaustive (NEX)**. For example, in Fig. 1, the concepts "Professional" and "Non_Professional" are EX, and the concept "Football" is NEX. Leaf nodes in the DAG are classified into the two categories of **no binding (NB)** and **run-time binding (RB)**. For NB, the concept's value is empty. In other words, no instantiation is required. In contrast, for RB, the value is instantiated on the fly during annotation or user query. We deploy more specific concepts for annotation and discard their corresponding generalized concepts. Leaf concepts and NEX concepts of the ontology only serve as metadata for audio objects. For NB, only tags and, for RB, both tags and values of concepts are stored in the database. Further, for NEX concepts, only tags can be stored.

3. INFORMATION SELECTION

The basic intuition for the generation of SQL is: first, users requests are matched with concepts of the ontology. Second, the SQL is written accordingly using the matched concepts. It is important to note that the more specific concepts of these matched concepts' are used in SQL generation. Further, virtually by residing on top of the database, ontology facilitates users more flexibility and abstraction to ask queries into the system.

To describe the user selection requests, the sample database is described with the following self-explanatory schema: **Audio_News (Id**, Time_Start, Time_End, Meta_News, ...). Note that the Meta_News attribute corresponds to a set of tags and values. We assume Meta_News to be expressed as a set-valued attribute which is a feature of object-relational DBMS (SQL3). The basic intuition for the generation of query in SQL is as follows: tokens are generated from the user's requested text after stemming. Tokens are associated with concepts when the synonymous lists of these concepts are matched with tokens. We call each of these associated concepts a **QConcept**. We check whether this QConcept is a leaf concept. If it is already a leaf concept, we determine this leaf concept's category. If it is an NB type, its tag is only added in the "where" clause. For an RB type, the value is instantiated and tag:value pair is added to the "where" clause. However, if this QConcept is a non-leaf concept, all of its children concepts are generated using depth/breadth first search (DFS/BFS). When children concepts become leaf concepts, these leaf concepts' tags or tag:value pairs are added to the "where" clause connected via boolean "or" condition. The following example illustrates the above notion: the user request, "Please give me news about the national football league." Since "NFL" turns out to be the QConcept which is itself a leaf concept with an NB type. Hence, the Query in SQL:

SELECT Time_Start, Time_End
FROM Audio_News
WHERE "NFL" **in** Meta_News

Let us consider the user request, "Please give me news about football." Note that the concept "Football" has four leaf concepts ("NFL," "CFL," "College_Football," and "HighSchool _Football " in Fig. 1) and the concept "Football" is NEX. In SQL:

SELECT Time_Start, Time_End
FROM Audio_News
WHERE "NFL" **in** Meta_News **or** "CFL" **in** Meta_News
or "College_Football" **in** Meta_News
or"HighSchool_Football" **in** Meta_News
or "Football" **in** Meta_News

4. EXPERIMENTAL IMPLEMENTATION

The development of PAC is still in progress. However, from its current state, we can report on the foundation of our initial choices (Fig. 2). In particular, our proposed framework follows a client-server architecture with modules written in Java. An object-relational DBMS (i.e., Informix Universal Server) is used to provide the functionality of the database. After getting sports content from the *Los Angeles Times*, audio feeds are stored in the server. Total duration of stored audio is 5 hours and 94 audio objects are defined. Most of these objects are annotated with single specific concept. To access data from the remote database to the client, Remote Method Invocation (*RMI*--a part of Java API provided by Informix) is used. At the client side, user input interface facilitates user profile information acquisition and voice input. User explicit and implicit feedback is stored at the server. To recognize voice input and convert into text, we use a speech recognition engine, IBM Via Voice and Java Speech API. The SQL generation is done by a module, which we call the SQL generator (SG). Then another module, which we call the query rewriter (QR), rewrites the query to achieve optimization.

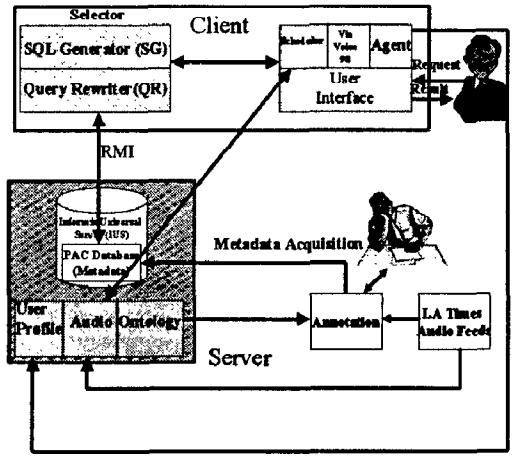

Figure 2. System Architecture of PAC

5. ACKNOWLEDGMENTS

Special thanks to my advisor, Professor Dennis Mcleod for his constructive and valuable suggestions on this work. This research has been funded in part by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center with additional support from the Annenberg Center for Communication at the University of Southern California and the California Trade and Commerce Agency.