The VLDB Journal (2004) 13: 71–85 / Digital Object Identifier (DOI) 10.1007/s00778-003-0105-1

Retrieval effectiveness of an ontology-based model for information selection

Latifur Khan[1], **Dennis McLeod**[2], **Eduard Hovy**[3]

[1] Department of Computer Science, University of Texas at Dallas, Richardson, TX 75083-0688, USA
e-mail: lkhan@utdallas.edu
[2] Department of Computer Science, University of Southern California, Los Angeles, CA 90088, USA
e-mail: mcleod@usc.edu
[3] Information Sciences Institute, University of Southern California, Marina del Rey, CA 90292, USA
e-mail: hovy@isi.edu

Edited by F. Lochovsky. Received: October 7, 2002 / Accepted: May 20, 2003
Published online: September 30, 2003 –

Abstract. Technology in the field of digital media generates huge amounts of nontextual information, audio, video, and images, along with more familiar textual information. The potential for exchange and retrieval of information is vast and daunting. The key problem in achieving efficient and user-friendly retrieval is the development of a search mechanism to guarantee delivery of minimal irrelevant information (high precision) while insuring relevant information is not overlooked (high recall). The traditional solution employs keyword-based search. The only documents retrieved are those containing user-specified keywords. But many documents convey desired semantic information without containing these keywords. This limitation is frequently addressed through query expansion mechanisms based on the statistical co-occurrence of terms. Recall is increased, but at the expense of deteriorating precision.

One can overcome this problem by indexing documents according to context and meaning rather than keywords, although this requires a method of converting words to meanings and the creation of a meaning-based index structure. We have solved the problem of an index structure through the design and implementation of a concept-based model using domain-dependent ontologies. An ontology is a collection of concepts and their interrelationships that provide an abstract view of an application domain. With regard to converting words to meaning, the key issue is to identify appropriate concepts that both describe and identify documents as well as language employed in user requests. This paper describes an automatic mechanism for selecting these concepts. An important novelty is a scalable disambiguation algorithm that prunes irrelevant concepts and allows relevant ones to associate with documents and participate in query generation. We also propose an automatic query expansion mechanism that deals with user requests expressed in natural language. This mechanism generates database queries with appropriate and relevant expansion through knowledge encoded in ontology form.

Focusing on audio data, we have constructed a demonstration prototype. We have experimentally and analytically shown that our model, compared to keyword search, achieves a significantly higher degree of precision and recall. The techniques employed can be applied to the problem of information selection in all media types.

Keywords: Metadata – Ontology – Audio – SQL – Precision – Recall

1 Introduction

The ever-increasing amount of useful multimedia information being created (e.g., on the Web) requires techniques for effective search and retrieval. Nontextual information, such as audio, video, and images, as well as more familiar textual information must be accommodated. A key problem is that users can be easily overwhelmed by the amount of information available via electronic means. The transfer of irrelevant information in the form of documents (e.g., text, audio, video) retrieved by an information retrieval system and that are of no use to the user wastes network bandwidth and frustrates users. This condition is a result of inaccuracies in the representation of the documents in the database as well as confusion and imprecision in user queries since users are frequently unable to express their needs efficiently and accurately. These factors contribute to the loss of information and to the provision of irrelevant information. Therefore, the key problem to be addressed in information selection is the development of a search mechanism that will guarantee the delivery of a minimum of irrelevant information (high precision) as well as insuring that relevant information is not overlooked (high recall).

The traditional solution to the problem of recall and precision in information retrieval employs keyword-based search techniques. Documents are retrieved if they contain (some combination of the) keywords specified by the user. However, many documents contain the desired semantic information, even though they do not contain the user-specified keywords.

This research has been funded [or funded in part] by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152.

This limitation can be addressed through the use of a query expansion mechanism. Additional search terms are added to the original query based on the statistical co-occurrence of terms [24]. Recall will be increased, but generally at the expense of deteriorating precision [21,29]. In order to overcome the shortcomings of keyword-based techniques in responding to information selection requests, we have designed and implemented a concept-based model using ontologies [14]. This model, which employs a domain-dependent ontology, is presented in this paper. An ontology is a collection of concepts and their interrelationships that can collectively provide an abstract view of an application domain [6,9].

There are two key problems in using an ontology-based model: one is the extraction of the semantic concepts from the keywords, and the other is the document indexing. With regard to the first problem, the key issue is to identify appropriate concepts that describe and identify documents on the one hand, and on the other the language employed in user requests. In this, it is important to make sure that irrelevant concepts will not be associated and matched and that relevant concepts will not be discarded. In other words, it is important to insure that high precision and high recall will be preserved during concept selection for documents or user requests. In this paper, we propose an automatic mechanism for the selection of these concepts from user requests, addressing the first problem. This mechanism will prune irrelevant concepts while allowing relevant concepts to become associated with user requests. Furthermore, a novel, scalable disambiguation algorithm for concept selection from documents using domain specific ontology is presented in [15].

With regard to the second problem, document indexing, one can use a vector space model of concepts or a richer and more precise method employing an ontology. We adopt the latter approach. A key reason for our choice is that the vector space model does not work well for short queries. Furthermore, a recent survey on Web search engines suggests that the average length of user requests is 2.2 keywords [5]. To address this, we have developed a concept-based model, which uses domain-dependent ontologies for responding to information selection requests. To improve retrieval precision, we also propose an automatic query expansion mechanism that deals with user requests expressed in natural language. This automatic expansion mechanism generates database queries by allowing only appropriate and relevant expansion. Intuitively, to improve recall during the phase of query expansion, only controlled and correct expansion is employed, guaranteeing that precision will not be degraded as a result of this process. Further, for the disambiguation of concepts, only the most appropriate concepts are selected with reference to documents or to user requests by taking into account the encoded knowledge in the ontology.

In order to demonstrate the effectiveness of our disambiguation model, we have explored and provided a specific solution to the problem of retrieving audio information. The effective selection/retrieval of audio information entails several tasks, such as metadata generation (description of audio), and the consequent selection of audio information in response to a query. Relevant to our purpose, ontologies can be fruitfully employed to facilitate metadata generation. For metadata generation, we need to do content extraction by relying on speech recognition technology that converts speech to text. After generating transcripts, we can deploy our ontology-based model to facilitate information selection requests. At present, an experimental prototype of the model has been developed and implemented. As of today, our working ontology has around 7000 concepts for the sports news domain, with 2481 audio clips/objects in the database. For sample audio content we use CNN broadcast sports and Fox Sports audio, along with closed captions. To illustrate the power of ontology-based over keyword-based search techniques, we have taken the most widely used vector space model as representative of keyword search. For comparison metrics, we have used measures of precision and recall and an F score that is the harmonic mean of precision and recall. Nine sample queries were run based on the categories of broader query (generic), narrow query (specific), and context query formulation. We have observed that on average our ontology outperforms keyword-based techniques. For broader and context queries, the result is more pronounced than in cases of narrow queries.

The remainder of this paper is organized as follows. In Sect. 2, we review related work. In Sect. 3, we introduce the research context in terms of the information media used (i.e., audio) and some related issues that arise in this context. In Sect. 4, we introduce our domain-dependent ontology. In Sect. 5, we present metadata management issues that arise for our ontology-based model in the context of audio information. In Sect. 6, we present a framework through which user requests expressed in natural language can be mapped into database queries via our ontology-based index structure, along with a pruning algorithm. In Sect. 7, we give a detailed description of the prototype of our system and provide data showing how our ontology-based model compares with traditional keyword-based search techniques. Finally, in Sect. 8 we present our conclusions and plans for future work.

2 Related work

Historically, ontologies have been employed to achieve better precision and recall in text retrieval systems [10]. Here, attempts have taken two directions – query expansion through the use of semantically related terms and the use of conceptual distance measures, as in our model. Among attempts using semantically related terms, query expansion with a generic ontology – WordNet [18] – has been shown to be potentially relevant to enhanced recall as it permits matching a query to relevant documents that do not contain any of the original query terms. Voorhees [27] manually expands 50 queries over a TREC-1 collection using WordNet and observes that expansion was useful for short, incomplete queries but not promising for complete topic statements. Further, for short queries automatic expansion is not trivial; it may degrade rather than enhance retrieval performance. This is because WordNet is too incomplete to model a domain sufficiently. Furthermore, for short queries less context is available, which makes the query vague. Therefore, it is difficult to choose appropriate concepts automatically.

The notion of conceptual distance between query and document provides an alternative approach to modeling relevance. Smeaton et al. [24] and Gonzalo et al. [8] focus on managing short and long documents, respectively. Note here that in these approaches queries and document terms are manually disambiguated using WordNet. In our case, query expansion and the selection of concepts, along with the use of the pruning algorithm, are fully automatic.

Although we use audio, here we show related work in the video domain that is closest to and complements our approach in the context of data modeling for the facilitation of information selection requests. Key related work in the video domain for selection of video segments includes [1,13,20]. Of these, Omoto et al. [20] use a knowledge hierarchy to facilitate annotation, while others use simple keyword-based techniques without a hierarchy. The model of Omoto et al. fails to provide a mechanism that automatically converts a generalized description into a specialized one(s). Further, this annotation is manual and does not deal with the disambiguation issues related to concepts.

3 Research context: audio

Audio is one of the most powerful and expressive of the non-textual media. Audio is a streaming medium (temporally extended), and its properties make it a popular medium for capturing and presenting information. At the same time, these very properties, along with audio's opaque relationship to computers, present several technical challenges from the perspective of data management [7]. The type of audio considered here is broadcast audio. In general, within a broadcast audio stream, some items are of interest to the user and some are not. Therefore, we need to identify the boundaries of news items of interest so that these segments can be directly and efficiently retrieved in response to a user query. After segmentation, in order to retrieve a set of segments that match with a user request, we need to specify the content of segments. This can be achieved using content extraction through speech recognition. Therefore, we present segmentation and content extraction techniques one by one.

3.1 Segmentation of audio

Since audio is by nature totally serial, arbitrary random access to audio information may be of limited use. To facilitate access to useful segments of audio information within an audio recording deemed relevant by a user, we need to identify entry points/jump locations. Further, multiple contiguous segments may form a relevant and useful news item.

As a starting point, both a change of speaker and long pauses can serve to identify entry points [3]. For long pause detection, we use short-time energy (En), which provides a measurement for distinguishing speech from silence for a frame (consisting of a fixed number of samples), which can be calculated by the following equation [22]:

$$En = \sum_{m=-\infty}^{m=\infty} [x(m)w(n-m)]^2 = \sum_{m=n-N+1}^{m=n} x(m)^2$$

where $x(m)$ is discrete audio signals, n is the index of the short-time energy, and $w(m)$ is a rectangle window of length N. When the En falls below a certain threshold, we treat this frame as a pause. After such a pause has been detected, we can combine several adjacent pauses and identify what can be called a long pause. Therefore, the presence of speeches with starting and ending points defined in terms of long pauses allows us to detect the boundaries of audio segments.

3.2 Content extraction

To specify the content of media objects, two main approaches have been employed: fully automated content extraction [11] and selected content extraction [28]. In fully automated content extraction, speech is converted to equivalent text (e.g., Informedia). Word-spotting techniques can provide selected content extraction in a manner that will make the content extraction process automatic. Word spotting is a particular application of automatic speech recognition techniques in which the vocabulary of interest is relatively small. In our case, vocabularies of concepts from the ontology can be used. Furthermore, content description can be provided in plain text, such as closed captions. However, this manual annotation is labor intensive. For content extraction, we rely on closed captions that came with an audio object itself from Fox Sports and the CNN Web site in our case (see Sect. 7).

3.3 Definition of an audio object

An audio object, by definition and in practice, is composed of a sequence of contiguous segments. Thus in our model the start time of the first segment and the end time of the last segment of these contiguous segments are used respectively to denote start time and end time of the audio object. Further, in our model pauses between interior segments are kept intact in order to insure that speech will be intelligible. The formal definition of an audio object indicates that an audio object's description is provided by a set of self-explanatory tags or labels using ontologies. An audio object O_i is defined by five tuple $(id_i, S_i, E_i, V_i, A_i)$, where Id_i is an object identifier that is unique, S_i is the start time, E_i is the end time, V_i (description) is a finite set of tags or labels, i.e., $V_i = \{v_{1i}, v_{2i}, \ldots, v_{ji}, \ldots, v_{ni}\}$ for a particular j where v_{ji} is a tag or label name, and A_i is simply audio recording for that time period. For example, an audio object is defined as $\{$10, 1145.59, 1356.00, $\{$Gretzky Wayne$\}$, *$\}$. Of the information in the five tuple, the first four items (identifier, start time, end time, and description) are called *metadata*.

4 Ontologies

An ontology is a specification of an abstract, simplified view of the world that we wish to represent for some purpose [6,9]. Therefore, an ontology defines a set of representational terms that we call *concepts*. Interrelationships among these concepts describe a target world. An ontology can be constructed in two ways, domain dependent and generic. CYC [17], WordNet [18], and Sensus [25] are examples of generic ontologies. For our purposes, we choose a domain-dependent ontology.

This is because, first, a domain-dependent ontology provides concepts in a fine grain, while generic ontologies provide concepts in coarser grain. Second, a generic ontology provides a large number of concepts that may contribute to a larger speech recognition error.

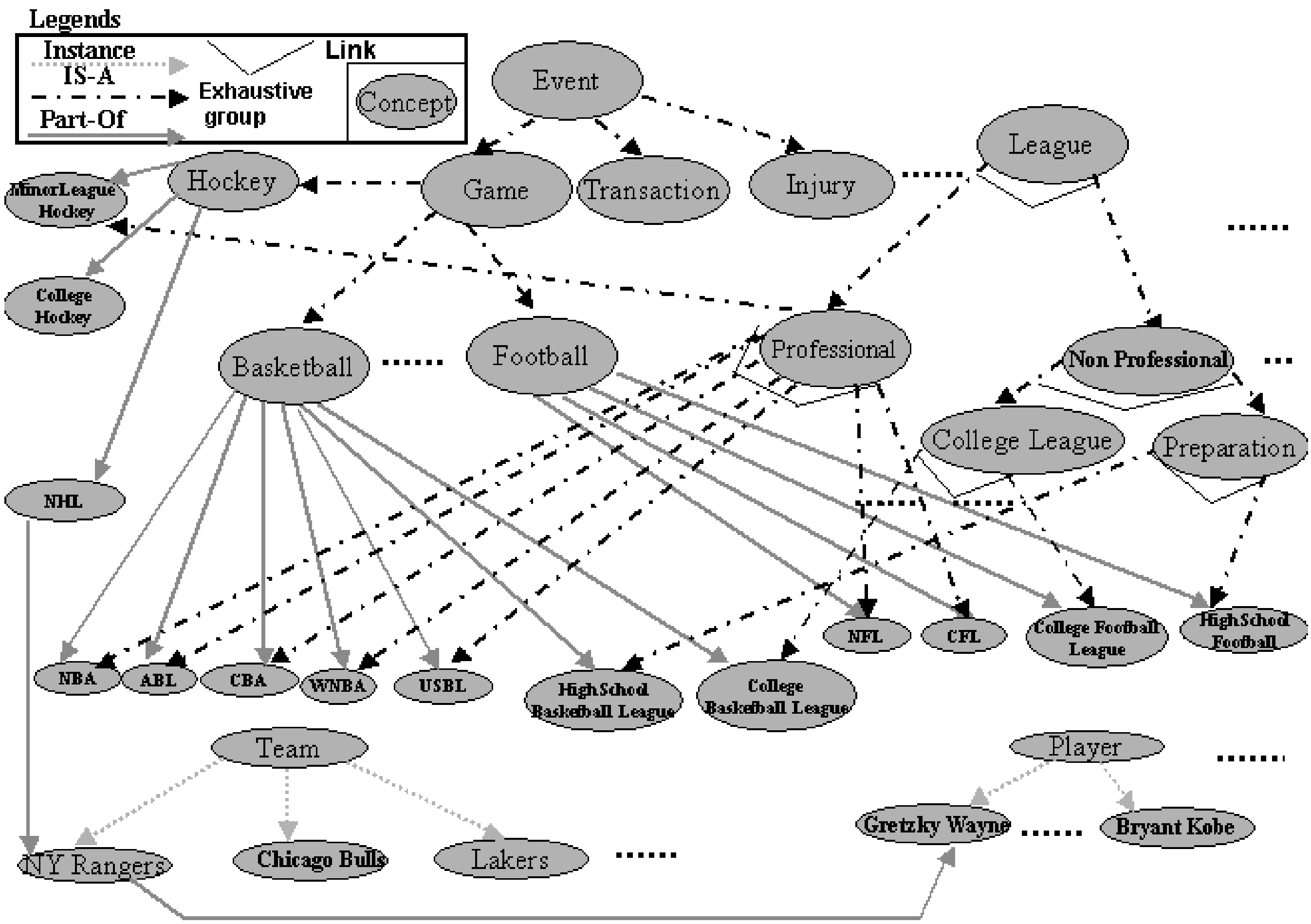

Fig. 1. A small portion of an ontology for a sports domain

Figure 1 shows an example ontology for sports news. Such an ontology is usually obtained from generic sports terminology and domain experts. Here we represent our ontology as a directed acyclic graph (DAG). Each node in the DAG represents a concept. In general, each concept in the ontology contains a label name and a synonyms list. Note also that this label name is unique in the ontology. Further, this label name is used to serve in associations of concepts with audio objects. The synonyms list of a concept contains vocabulary (a set of keywords) through which the concept can be matched with user requests. Formally, each concept has a synonyms list $(l_1, l_2, l_3, \ldots, l_i, \ldots, l_n)$ where user requests are matched with this l_i – an *element* of the list. Note that a keyword may be shared by multiple concepts' synonyms lists. For example, player "Bryant Kobe", "Bryant Mark", and "Reeves Bryant" share the common word "Bryant", which may of course create an ambiguity problem.

4.1 Interrelationships

Concepts are interconnected by means of interrelationships. If there is an interrelationship R between concepts C_i and C_j, then there is also an interrelationship $R\prime$ (R inverse) between concepts C_j and C_i. In Fig. 1, interrelationships are represented by labeled arcs/links. Three kinds of interrelationships are used to create our ontology: specialization (IS-A), instantiation (Instance-Of), and component membership (Part-Of). These correspond to key abstraction primitives in typical object-based and semantic data models [4].

IS-A: This interrelationship is used to represent specialization (concept inclusion). A concept represented by C_j is said to be a specialization of the concept represented by C_i if C_j is a kind of C_i. For example, "NFL" is a kind of "Professional" league. In other words, "Professional" league is the generalization of "NFL". In Fig. 1, the IS-A interrelationship between C_i and C_j goes from generic concept C_i to specific concept C_j, represented by a broken line. IS-A interrelationships can be further categorized into two types: *exhaustive* and *nonexhaustive*. An exhaustive group consists of a number of IS-A interrelationships between a generalized concept and a set of specialized concepts and places the generalized concept into a categorical relation with a set of specialized concepts in such a way that the union of these specialized concepts is equal to the generalized concept. For example, "Professional" relates to a set of concepts, "NBA", "ABL", "CBA", . . . , by exhaustive group (denoted by caps in Fig. 1). Further, when a generalized concept is associated with a set of specific concepts by only IS-A interrelationships that fall into the exhaustive group, then this generalized concept will not participate explicitly in the metadata generation and SQL query generation. This is because this generalized concept is entirely partitioned into its specialized concepts through an exhaustive group. We call this generalized concept a *nonparticipant concept (NPC)*. For example, in Fig. 1 "Professional" concept is NPC. On the other hand, a nonexhaustive group consisting of a set of IS-A does not exhaustively categorize a generalized concept into a set of specialized concepts. In other words, the union of specialized concepts is not equal to the generalized concept.

Instance-Of: The Instance-Of relationship denotes concept instantiation. If a concept C_j is an example of concept C_i, the interrelationship between them corresponds to an Instance-Of denoted by a dotted line. For example, player "Wayne Gretzky" is an instance of the concept "Player". In general, all players and teams are instances of the concepts "Player" and "Team", respectively.

Part-Of: A concept is represented by C_j is Part-Of a concept represented by C_i if C_i has a C_j (as a part) or C_j is a part of C_i. For example, the concept "NFL" is Part-Of the concept "Football" and player, and "Wayne Gretzky" is Part-Of the concept "NY Rangers".

4.2 Disjunct concept

When a number of concepts are associated with a parent concept through IS-A interrelationships, it is important to note when these concepts are disjoint and are referred to as concepts of a disjoint type. When, for example, the concepts "NBA", "CBA", or "NFL" are associated with the parent concept "Professional" through IS-A, they become disjoint concepts. Moreover, any given object's metadata cannot possess more than one such concept of the disjoint type. For example, when an object's metadata are the concept "NBA", it cannot be associated with another disjoint concept such as "NFL". It is noteworthy that the property of being disjoint helps to disambiguate concepts for keywords in user request (query) disambiguation. Similarly, the concepts "College Football" and "College Basketball" are disjoint concepts due to their associations with the parent concept "College League" through IS-A. Furthermore, "Professional" and "Non Professional" are disjoint. Thus, we can say that "NBA", "CBA", "ABL", "College Basketball", and "College Football" are disjoint. Each of these leagues and its team and players form a boundary that form a *region*.

During annotation of concepts with an audio object we strive to choose a particular region. This is because an audio object can be associated with only one disjoint-type concept. However, it may be possible that a particular player plays in several leagues. In that case, we consider two alternatives. First, we may generate multiple instances of the player in the ontology. In other words, for each league in which the player plays he will be represented by a separate concept. In this manner, we are able to preserve the property of disjunction. In this case, each region is simply a subtree. Second, we may keep just one node for the player that has two parents, say, two teams. In this case, each region is DAG. With the former approach, maintenance or update will be an issue; inconsistency may arise. With the latter approach, maintenance will be easier; however, precision will be hurt. This is because if the query is requested in terms of a team where this player plays, some retrieved objects will be related to the other team and vice versa. This is because both teams have a common child concept, and the query expansion phase allows one to retrieve all associated audio objects to this player regardless of his teams. For example, the player "Deion Sanders" plays two teams, "Dallas Cowboys" (under NFL region) and Cincinnati Reds (under MLB region). If the user request is specified by "Dallas Cowboys", some objects will be retrieved that contain information "Cincinnati Reds" along with "Deion Sanders". For this, we adopt the former approach.

On the other hand, concepts are not disjoint when they are associated with a parent concept through Instance-Of or Part-Of. In these cases, some of these concepts may serve simultaneously as metadata for an audio object. An example would be the case in which the metadata of an audio object are team "NY Rangers" and player "Gretzky Wayne", where "Gretzky Wayne" is Part-Of "NY Rangers".

4.3 Creating an ontology

For the construction of sports domain-dependent ontologies, first we list all possible objects necessary to cover a given sports domain. This possible object list should include different sports, such as basketball, football, baseball, hockey etc., and different leagues within a given sport, as in basketball, NBA, ABL, CBA, and so forth. Furthermore, different sports can be qualified by characteristics such as injuries, player transactions, strikes, etc.

Now the question becomes what level of granularity of knowledge we need to take into account in the ontology. Since our goal is to build a search mechanism that is more powerful than keyword-based search techniques without relying upon the understanding of natural language, we do not, in our ontology, represent concepts at the level of granularity necessary, for example, for the extraction of highly specific factual information from documents (e.g., how many times a batter attempted to execute a hit-and-run play in a particular game). Yet the term "Transaction" has been further qualified through the use of such designations, within a given sports domain, as "sign a player", "retire a player", "release a player", "trade a player", "draft a player", and so on. Furthermore, all instances of teams, players, and managers are grouped under "team", "player", and "manager", respectively, through Instance-Of interrelationships. Note that, specifically, a set of players who play in a team are grouped into that team through Part-Of interrelationships. All teams, players, and managers for different leagues are taken from Yahoo. Note that Yahoo has a hierarchy of 151,763 categories [16]. Only a part of this hierarchy has been used for the construction of our domain-dependent ontology. Therefore, upper-level concepts are chosen manually; however, lower-level concepts are borrowed from the Yahoo hierarchy. The sports subcategory (Recreation category) of Yahoo's hierarchy is considered here. Only teams and players along with different leagues under the sports category from Yahoo's hierarchy are added in the ontology. The maximum depth of our ontology's DAG is six. The maximum number of children concepts from a concept is 28 (branching factor).

Now the question is: how much does this ontology cover? What are the criteria for the test model? Since lower-level concepts are taken from the Yahoo hierarchy and more upper-level concepts are added in the ontology, we believe that the ontology covers the domain reasonably well. Yahoo's inclusion provides direct practical evidence that these concepts have been found useful by people. In addition, we have conducted some experiments in order to estimate coverage. In these experiments, our goal is to select concepts from ontologies for the annotated text of audio clips. These clips are taken from the CNN sports and the Fox Sports Web site along with their closed captions, not from the Yahoo Web site. We have observed that 90.5% of the clips are associated with concepts of ontologies, while 9.5% of the clips failed to associate with any concept of ontologies due to incompleteness of ontologies. One important observation about our ontology is that it is constructed from the perspective of a database. Therefore, in the ontology the mechanism of inference sometimes may not be supported among concepts for different links or arcs.

5 Metadata acquisition and management of metadata

Metadata acquisition is the name of the process through which descriptions are provided for audio objects. For each audio object, we need to find the most appropriate concept(s). Recall that by using content extraction (Sect. 3.2) we get a set of keywords that appear in a given audio object. For this, concepts from ontologies will be selected based on matching terms taken from their lists of synonyms with those based on specified keywords. Furthermore, each of these selected concepts will have a score based on a partial or a full match. It is possible that a particular keyword may be associated with more than one concept in the ontology. In other words, the association between keyword and concept is one:many, rather than one:one. Therefore, the disambiguation of concepts is required. The basic notion of disambiguation is that a set of keywords occurring together determines a context for one another, according to which the appropriate senses of the word (its appropriate concept) can be determined. Note, for example, that base, bat, glove may have several interpretations as individual terms, but when taken together the intent is obviously a reference to baseball. The reference follows from the ability to determine a context for all the terms. Thus extending and formalizing the idea of context in order to achieve the disambiguation of concepts, we propose an efficient pruning algorithm based on two principles: co-occurrence and semantic closeness. This disambiguation algorithm first strives to disambiguate across several regions using the first principle and then disambiguates within a particular region using the second (see [15] for more details).

Effective management of metadata facilitates efficient storing and retrieval of audio information. To this end, in our model the most specific concepts are considered as metadata. Several concepts of the ontology, for example, can be candidates for the metadata of an audio object. However, some of these may be children of others. Two alternative approaches can be used to address this problem. First, we can simply store the most general concepts. But we may get many irrelevant objects (precision will be hurt) for queries related to specific concepts. For example, an audio object becomes the candidate for the concepts "NHL", "Hockey", and "Professional". We can simply store the general concept, "Professional", for this object. When a user request comes in terms of a specific concept, "NHL", this object will be retrieved along with other irrelevant objects that do not belong to NHL (say, NFL, CFL, and so on). Therefore, precision will be hurt. Second, the most specific concepts can be stored in the database. Corresponding generalized concepts can then be discarded. In this case, recall will be hurt. Suppose, for example, an audio object becomes the candidate for the concepts "NHL", "Hockey", and "Professional". During the annotation process, the object will only be annotated with the most specific concept, "NHL". In this case, the metadata of the audio objects stored in the database will be comprised of the most specific concepts. If the query comes in terms of "Hockey" or "Professional", this object will not be retrieved.

We follow the latter approach. By storing as metadata specific concepts, rather than generalized concepts of the ontology, we can expect to achieve the effective management of metadata. In order to avoid recall problems, user requests are first passed through the ontology on the fly and expressed in terms of the most specific concepts. Even so, the audio object in the above example can still be retrieved through querying the system by "NHL", "Hockey", and "Professional".

Here we consider an efficient way of storing audio objects in the database: we maintain a single copy of all the audio data in the database. Further, each object's metadata are stored in the database. Thus the start time and end time of an object point to a fraction of all the audio data. Therefore, when the object is selected, this boundary information provides relevant audio data that are to be fetched from all the audio data and played by the scheduler. The following self-explanatory schemas are used to store audio objects in the database: *Audio_News (Id, Time_Start, Time_End, . . .), and Meta_News (Id, Label)*. Each audio object's start time, end time, and description correspond to Time_Start, Time_End, and Label, respectively. Furthermore, each object's description is stored as a set of rows or tuples in the Meta_News table for normalization purposes.

6 Query mechanisms

We now focus specifically on our techniques for utilizing an ontology-based model for processing information selection requests. In our model, the structure of the ontology facilitates indexing. In other words, the ontology provides index terms/concepts that can be used to match with user requests. Furthermore, the generation of a database query takes place after the keywords in the user request are matched to concepts in the ontology.

We assume that user requests are expressed in plain English. Tokens are generated from the text of the user's request after stemming and removing stop words. Using a list of synonyms, these tokens are associated with concepts in the ontology through Depth First Search (DFS) or Breadth First Search (BFS). Each of these selected concepts is called a *QConcept*. Among QConcepts, some might be ambiguous. However, through the application of a pruning technique that will be discussed in Sect. 6.1, only relevant concepts are retained. These relevant concepts will then be expanded and will participate in SQL query generation, as is discussed in Sect. 6.2.

6.1 Pruning

Disambiguation is needed when a given keyword matches more than one concept. In other words, multiple concepts will have been selected for a particular keyword. For disambiguation, it is necessary to determine the correlation between the selected concepts based on semantic closeness. When concepts are correlated, the scores of concepts strongly associated with each other will be given greater weight based on their minimal distance from each other in the ontology and their own matching scores based on the number of words they match. Thus, ambiguous concepts which correlate with other selected concepts will have a higher score, and a greater probability of being retained than ambiguous concepts which are not correlated.

For example, if a query is specified by "Please tell me about team Lakers", QConcepts "Team", "Los Angeles Lakers" (an NBA league team), and major league baseball player "Tim Laker" (of team "Pittsburgh Pirates") are selected. Note that the selected concepts "Los Angeles Lakers" and "Tim Laker" are ambiguous. However, "Los Angeles Lakers" is associated with the selected QConcept "Team" by the Instance-Of interrelationship, while "Tim Laker" is not directly related to anything selected. Therefore, we prune the noncorrelated ambiguous concept, player "Tim Laker". The above idea is implemented using score-based techniques. Now we would like to present our concept-pruning algorithm for use with user requests.

6.1.1 Formal definitions

Each selected concept contains a score based on the number of keywords from the list of synonyms that have been matched with the user request. Recall that in an ontology, each concept (QCi) has a complementary list of synonyms (l1, l2, l3, ..., lj ,..., ln). Keywords in the user request are sought that match each keyword on the element lj of a concept. The calculation of the score for lj, which we designate an Escore, is based on the number of matched keywords of lj. The largest of these scores is chosen as the score for this concept and designated Score. Furthermore, when two concepts are correlated, their scores, called the propagated-score, are inversely related to their position (semantic distance) in the ontology. Let us formally define each of these scores.

Definition 1. Element-score (Escore): *The Element-score of an element l_j for a particular QConcept QC_i is the number of keywords of l_j matched with keywords in the user request divided by the total number of keywords in l_j.*

$$Escoreij \equiv \frac{\#\ \textit{of keywords of}\ lj\textit{matched}}{\parallel \#\textit{of keywords in}\ lj \parallel}$$

The denominator is used to nullify the effect of the length of lj on $Escoreij$ and insures that the final weight is between 0 and 1.

Definition 2. Concept-score (Score): *The concept-score for a QConcept QC_i is the largest score of all its Escores. Thus,*

$$Scorei = \max Escoreij \quad \textit{where}\ 1 \le j \le n$$

Definition 3. Semantic distance $(SD(QC_i, QC_j))$: *$SD(QC_i, QC_j)$ between QConcepts QC_i and QC_j is defined as the shortest path between them in the ontology.* Note that if concepts are at the same level and no path exists, the semantic distance is infinite. For example, the semantic distance (SD) between concepts "NBA" and team "Lakers" is 1 (see Fig. 2). This is because the two concepts are directly connected via a Part-Of interrelationship. Similarly, the semantic distance between "NBA" and "Bryant Kobe" is 2. The SD between "Los Angeles Lakers" and "New Jersey Nets" is infinite.

Definition 4. Propagated-score (S_i): *If a QConcept, QC_i, is correlated with a set of QConcepts $(C_j, C_{j+1}, \ldots, C_n)$, the propagated-score of QC_i is its own Score, $Score_i$, plus the scores of each of the correlated QConcepts' $(QC_k k = j, j+1, \ldots, n) Score_k$ divided by $SD(QC_i, QC_k)$. Thus,*

$$\begin{aligned} Si &= Scorei + \sum_{k=j}^{k=n} \frac{Scorek}{SD(QCi, QCk)} \\ &= Scorei + \frac{Scorej}{SD(QCi, QCj)} \\ &\quad + \frac{Scorej+1}{SD(QCi, QCj+1)} + \ldots + \frac{Scoren}{SD(QCi, QCn)} \end{aligned}$$

For example, in Fig. 2 let us assume that values of $Score_i$ for "Los Angeles Lakers" and "Bryant Kobe" are 0.5 and 1.0, respectively. Furthermore, these concepts are correlated with a semantic distance of 1, and their propagated-scores are 1.5 $(0.5 + 1.0/1)$ and 1.5 $(1.0 + 0.5/1)$, respectively. The pseudocode for the pruning algorithm is as shown in Fig. 3.

Using this pruning algorithm (see Fig. 3), for the user request "Team Lakers", the initially selected QConcepts are "Team", "Los Angeles Lakers", and "Tim Laker" (see Fig. 2). Note that "Laker" is ambiguous since it has concepts "Los Angeles Lakers" and "Tim Laker". In Fig. 2, the SD between concepts "Team" and "Los Angeles Lakers" is 1, while the SD between concepts "Team" and "Tim Laker" is 2. Furthermore, the *Scores* for concepts "Team", "Los Angeles Lakers", and "Tim Laker" are 1.0, 0.5, and 0.5, respectively. It is important to note that when two concepts are correlated with each other where SD is greater than one, they will have lower $propagated - scores\ S_i$ and S_j compared to concepts with the same concept-scores and a SD of 1. This is because for the higher semantic distance concepts are correlated in a broader sense. Thus, concepts that are correlated have a higher S_i in comparison with noncorrelated concepts. Now, the propagated-score for QConcepts "Team", "Los Angeles Lakers", and "Tim Laker" becomes 1.75 (1.0+0.5/1+0.5/2), 1.5 (0.5+1.0/1), and 1.0 (0.5+1.0/2), respectively. Therefore, we keep the concept "Los Angeles Lakers" from among these ambiguous concepts and prune the other. Thus, the SD helps us to discriminate between ambiguous concepts.

Among selected concepts, one concept may subsume other concepts. In this case, we use the most specific concept for SQL generation. For example, if a user request is expressed as "Please tell me about Lakers' Bryant", the QConcepts team "Los Angeles Lakers" and players "Bryant Kobe", "Bryant Mark", "Reeves Bryant" are selected. Their concept-scores are 0.5, 0.5, 0.5, and 0.5, respectively. "Bryant" is ambiguous among the latter three concepts. However, among these selected concepts, only "Bryant Kobe" and "Los Angeles Lakers" are correlated with a semantic distance of 1 (see Fig. 2).

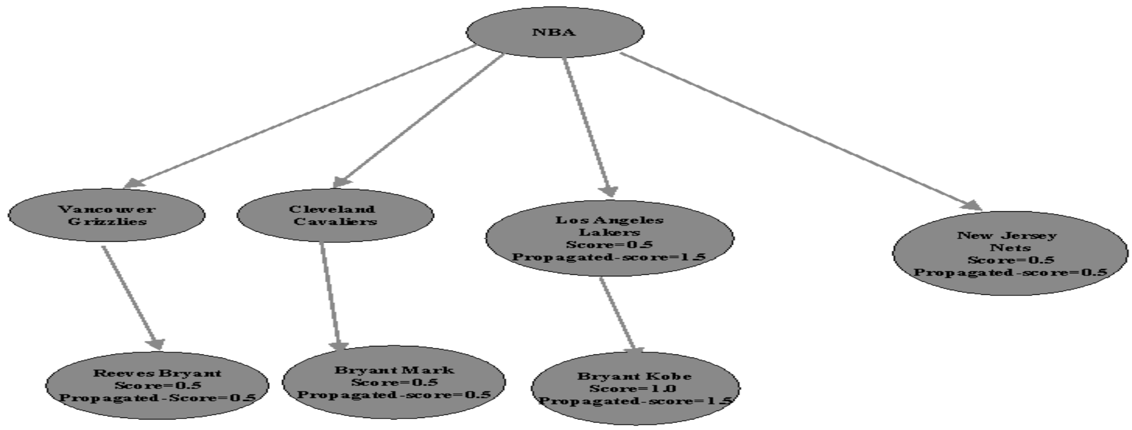

Fig. 2. Illustration of scores and propagated-scores of selected concepts

QC_1, QC_2, ..., QC_l, ..., QC_r are selected with concept-score $Score_1, \ldots, Score_l, \ldots Score_r$

Determine correlation of selected concepts $(QC_i, QC_j, QC_{j+1}, .., QC_n)$ and update their Propagated-scores using

$$S_i = Score_i + \sum_{k=j}^{k=n} \frac{Score_k}{SD(QC_i, QC_k)}$$

$$= Score_i + \frac{Score_j}{SD(QC_i, QC_j)} + \frac{Score_{j+1}}{SD(QC_i, QC_{j+1})} + \ldots + \frac{Score_n}{SD(QC_i, QC_n)}$$

Sort all QConcepts (QC_i) based on S_i in descending order

//Find ambiguous QConcepts and prune those which have low Propagated-score…

For a keyword that associated with several QConcepts, QC_i, QC_j, $QC_l, \ldots$ where $S_i > S_j > S_l$, ...

Keep only QC_i and discard QC_j, QC_l, …

//End of For Loop for a keyword.

Keep all specific QConcepts and discard corresponding generalized concepts

For each QConcept that is not pruned

Query_Expansion_SQL_Generation (QConcept)

//End of For loop each QConcept

Fig. 3. Pseudocode for pruning algorithm

Therefore, their propagated-scores S_i are higher compared to other concepts, in this case, 1.0, 1.0, 0.5, and 0.5, respectively. Consequently, we throw away "Bryant Reeves" and "Bryant Mark". Furthermore, "Bryant Kobe" is a subconcept of "Los Angels Lakers" due to a Part-Of interrelationship. In this case, we keep the more specific concept, "Bryant Kobe", and the SQL generation algorithm will be called for this QConcept only.

6.2 Query expansion and SQL query generation

We now discuss a technique for query expansion and SQL query generation. In response to a user request for the generation of a SQL query, we follow a Boolean retrieval model. We now consider how each QConcept is mapped into the "where" clause of a SQL query. Note that by setting the QConcept as a Boolean condition in the "where" clause, we are able to retrieve relevant audio objects. First, we check whether or not the QConcept is of the NPC type. Recall that NPC concepts can be expressed exhaustively as a collection of more specific concepts. If the QConcept is an NPC concept, it will not be added in the "where" clause. Otherwise, it will be added into the "where" clause. Likewise, if the concept is leaf node, no further progress will be made for this concept. However, if it is nonleaf node, its children concepts are generated using DFS/BFS, and this technique is applied for each children concept. One important observation is that all concepts appearing in a SQL query for a particular QConcept are expressed in disjunctive form. Furthermore, during the query expansion phase only correct concepts are added that will guarantee that the addition of new terms will not hurt precision. The complete algorithm is shown in Fig. 4.

The following example illustrates the above process. Suppose the user request is "Please give me news about player Kobe Bryant". "Bryant Kobe" turns out to be the QConcept that is itself a leaf concept. Hence the SQL query (for schema see Sect. 5) generated by using only "Bryant Kobe" (with the label "NBAPlayer9") is:

SELECT Time_Start, Time_End
FROM Audio_News a, Meta_News m
WHERE a.Id=m.Id
AND Label="NBAPlayer9"

Let us now consider the user request "Tell me about Los Angeles Lakers". Note that the concept "Los Angeles Lakers" is not of the NPC type, so its label ("NBATeam11") will be added in the "where" clause of the SQL query. Further, this concept has several children concepts ("Bryant Kobe", "Celestand John", "Horry Robert", . . . , i.e., names of players for this team). Note that these player concepts' labels are "NBAPlayer9", "NBAPlayer10", and "NBAPlayer11", respectively. In the SQL query:

SELECT Time_Start, Time_End
FROM Audio_News a, Meta_news m
WHERE a.Id = m.Id
AND (Label="NBATeam11"
OR Label="NBAPlayer9"
OR Label="NBAPlayer10". . .)

6.2.1 Remedy of explosion of Boolean condition

Since most specific concepts are used as metadata and our ontologies are large in the case of querying upper-level concepts, every relevant child concept will be mapped into the "where" clause of the SQL query and expressed as a disjunctive form. To avoid the explosion of Boolean conditions in this clause of the SQL query, the labels for the player and team concepts are chosen in an intelligent way. These labels begin with the label of the league in which the concepts belong. For example, team "Los Angeles Lakers" and player "Bryant, Kobe" are under "NBA". Thus the labels for these two concepts are "NBATeam11" and "NBAPlayer9", respectively, whereas the label for the concept "NBA" is "NBA".

Now, when user requests come in terms of an upper-level concept (e.g., "Please tell me about NBA"), the SQL query generation mechanism will take advantage of prefixing:

SELECT Time_Start, Time_End
FROM Audio_News a, Meta_News m
WHERE a.Id=m.Id
AND Label Like "%NBA%"

On the other hand, if we do not take advantage of prefixing, the concept "NBA" will be expanded into all its teams (28) and will let us assume each team has 14 players. Therefore, we need to maintain 421 (1+ 28 + 28 *14) Boolean conditions in the "where" clause of the SQL query. This explosion will be exemplified by upper-level concepts like basketball.

6.3 Query optimizations

The basic idea of query optimizations is to rewrite the database SQL query effectively in order to leverage the work of a traditional query optimizer.

6.3.1 Qualified disjunctive form (QDF)

When one concept qualifies another concept (e.g., professional football), the straightforward approach is to treat this as a conjunctive query. Then we may generate the query by simply writing two QConcepts as a Boolean "and" in the "where" clause. However, further optimization is possible by taking the intersection of all children concepts of the two QConcepts. Hence we consider only concepts that intersect for these QConcepts. By discarding the nonintersecting concepts, the number of Boolean conditions in the "where" clause for these QConcepts is reduced. However, the query (traditional) optimizer eliminates redundant children concepts by means of transforming a redundant expression into an equivalent one using Boolean algebra. By employing this technique, we leverage the work of a traditional query optimizer. Note that here for QConcepts that qualify one another all concepts in the "where" clause are expressed in disjunctive form. The query "give me professional football news" illustrates the conversion into QDF. The intersected concepts of two QConcepts (professional and football) are "NFL" and "CFL". The SQL for the optimized form is:

```
Query_Expansion_SQL_Generation (QC_i)
    Mark QC_i is already visited
If QC_i is not NPC Type
Add label of QC_i into where clause of SQL as
disjunctive form
//Regardless of NPC type concept
If QC_i is not leaf node and not visited yet
    For each children concept, QCh_l of QC_i
    using    DFS/BFS
     Query_Expansion_SQL_Genertaion (QCh_l )
```

Fig. 4. Pseudocode for SQL generation

SELECT Time_Start, Time_End
FROM Audio_News a, Meta_News m
WHERE a.Id=m.Id
AND ((Label Like "%NFL%")
OR (Label Like "%CFL%"))

6.3.2 Optimization for conjunctive queries

In a conjunctive query, the fact that some concepts are disjoint enables us to eliminate certain irrelevant concepts from the "where" cause of the SQL query. These are not necessarily a case in which the presence of these concepts will reduce precision; however, the query response time will be adversely affected. For example, if the user requests "tell me about the game between the Blazers and the Lakers", three Qconcepts – "Portland Trail Blazers", "Los Angeles Lakers", and "Tim Laker" (label "MLBPlayer111") – are selected. The last two, with regard to each other, are ambiguous concepts, called forth in response to the keyword "Laker". Note that "Los Angeles Lakers" and "Portland Trail Blazers" are two teams in the NBA, whereas "Tim Laker" is a Major League Baseball player. Since there is no correlation between "Los Angeles Lakers" and "Portland Trail Blazers", we are not led to throw away "Tim Laker" through the use of our disambiguation algorithm, which is discussed in Sect. 6.1.1.

Note that the concepts here, some of which are ambiguous, are distributed across several regions. In a conjunctive query, all QConcepts that are selected should be in the same region. Otherwise, they will return an empty set. This is because an audio object can only be associated with one region. Thus for concepts that are ambiguous for a particular keyword, only those will be retained that appear in a region in which overall the greatest number of concepts are selected in the case of a specific query (conjunctive). For this example, the ambiguous concepts "Los Angeles Lakers" and "Tim Laker" are in the region NBA and the region MLB, respectively, while the nonambiguous concept "Portland Trail Blazers" is in the region NBA. Thus, the concept "Tim Laker" will be thrown out. Thus ambiguous concepts in different regions are also automatically pruned. The SQL query is:

SELECT Time_Start, Time_End
FROM Audio_News a, Meta_news m1, Meta_news m2
WHERE a.Id = m1.Id **AND** a.Id=m2.Id
AND m1.Label="NBATeam11"
AND m2.Label="NBATeam21"

7 Experimental implementation

In discussing implementation, we will first present our experimental setup, and then we will demonstrate the power of our ontology-based over keyword-based search techniques. We have constructed an experimental prototype system based upon a client server architecture. The server (a SUN Sparc Ultra 2 model with 188 MB of main memory) has an Informix Universal Server (IUS), which is an object relational database system. For the sample audio content, we use CNN broadcast sports audio and Fox Sports. We have written a hunter program in Java that goes to these Web sites and downloads all audio and video clips with closed captions. The average size of the closed captions for each clip is 25 words, after removing stop words. These associated closed captions are used to hook with the ontology. As of today, our database has 2481 audio clips. The usual duration of a clip is not more than 5 min in length. Wav and ram are used for media format. Currently, our working ontology has around 7000 concepts for the sports domain. For fast retrieval, we load the upper-level concepts of the ontology in main memory, while leaf concepts are retrieved on a demand basis. Hashing is also used to increase the speed of retrieval.

7.1 Results

We would like to demonstrate the power of our ontology over keyword-based search techniques. For an example of a keyword-based technique we have used the most widely used model, the vector space model [23].

7.1.1 Vector space model

Here queries and documents are represented by vectors. Each vector contains a set of terms or words and their weights. The similarity between a query and a document is calculated based on the inner product or cosine of two vectors' weights. The weight of each term is then calculated based on the product of term frequency (TF) and inverse-document frequency (IDF). TF is calculated based on the number of times a term occurs in a given document or query. IDF is the measurement of inter-document frequency. Terms that appear unique to a document will have high IDF. Thus for N documents, if a term appears in n documents, the IDF for this term *=log(N/n) +1*. Let us assume query (Q_i) and document (D_j) have t terms and their associated weights are WQ_{ik} and WD_{ik}, respectively, for $k = 1$ to t. Similarity between these two is measured using the following inner product:

$$Sim(Qi, Dj) = Cosine(Qi, Dj) = \frac{\sum_{k=1}^{k=t} WQik * WDjk}{\sqrt{\sum_{k=1}^{k=t}(WQik)^2 * \sum_{k=1}^{k=t}(WDjk)^2}}$$

The denominator is used to nullify the effect of the length of the document and query and ensure that the final value is between 0 and 1.

7.1.1.1 Analytical results

Unlike standard database systems, such as relational or object-oriented systems, audio objects retrieved by a search technique are not necessarily of use to the user. This is due mainly to inaccuracies in the way these objects are presented in the interpretations of the audio objects and the users' queries and through the inability of users to express their retrieval needs precisely. A relevant object is an object of use to the user in response to his or her query, whereas an irrelevant object is one of little or no use. The effectiveness of retrieval is usually measured by the following two quantities, recall and precision:

$$Recall = \frac{\textit{Number of retrieved relevant objects}}{\textit{Number of relevant objects}}$$

$$Precision = \frac{\textit{Number of retrieved relevant objects}}{\textit{Number of retrieved objects}}$$

Let us assume that Rel represents the set of relevant objects and Ret represents the set of retrieved objects. The above measure can also be redefined in the following manner.

$$Recall = \frac{|Rel \cap Ret\ |}{|Rel\ |} \tag{1}$$

$$Precision = \frac{|Rel \cap Ret\ |}{|Ret\ |} \tag{2}$$

For example, if 10 documents are retrieved of which 7 are relevant, and the total number of relevant documents is 20, then recall $= 7/20$ and precision $= 7/10$.

Therefore, recall and precision denote, respectively, completeness of retrieval and purity of retrieval. A common phenomenon is that as recall increases, precision unfortunately decreases. This means that when it is necessary to retrieve more relevant audio objects a higher percentage of irrelevant objects will usually also be retrieved.

In order to evaluate the retrieval performance of two systems, we can employ an F score. The F score is the harmonic mean of recall and precision, a single measure that combines recall and precision. The function ensures that an F score will have values within the interval [0,1]. The F score is 0 when no relevant documents have been retrieved, and it is 1 when all retrieved documents are relevant. Furthermore, the harmonic mean F assumes a high value only when both precision and recall are high. Therefore, determination of the maximum value for F can be interpreted as an attempt to find the best possible compromise between recall and precision.

$$F\ score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3}$$

Let us assume that a user request is expressed by the single keyword W_1. A keyword-based search technique retrieves Ret_1, the set of objects where *Rel* is the set of relevant objects. Precision (P_k) and recall (R_k) for keyword-based search techniques are defined by:

$$R_k = \frac{|Rel \cap Ret_1\ |}{|Rel\ |} \tag{4}$$

$$P_k = \frac{|Rel \cap Ret_1\ |}{|Ret_1\ |} \tag{5}$$

Now we turn to the ontology-based model. Without loss of generality we assume that keyword W_1 chooses the NPC concept C_1 in the ontology, and for this concept a given set of objects, Ret_1, is retrieved. Furthermore, this concept can be expanded to include its subconcepts $C_2, C_3, C_4, C_5, \ldots, C_n$ for which the objects $Ret_2, Ret_3, Ret_4, Ret_5, \ldots, Ret_n$ are retrieved. Note that for purposes of simplification, we assume that concept C_1 and keyword W_1 each retrieve the same set of objects, Ret_1.

Precision (P_o) and recall (R_o) for the ontology-based search technique are defined by:

$$R_o = \frac{\sum_{i=1}^{n} |Rel \cap Ret_i\ |}{|Rel\ |} \tag{6}$$

$$P_o = \frac{\sum_{i=1}^{n} |Rel \cap Ret_i\ |}{\sum_{i=1}^{n} |Ret_i\ |} \tag{7}$$

Therefore, from Eqs. 4 and 6 we get

$$\frac{R_o}{R_k} = 1 + \frac{|\ Rel \cap Ret_2\ | + \ldots + |Rel \cap Ret\ n|}{|Rel\ \cap Ret_1|} \tag{8}$$

$$\frac{R_o}{R_k} \geq 1 \tag{9}$$

since

$$\frac{|\ Rel \cap Ret_2\ | + \ldots + |Rel \cap Ret_n\ |}{|Rel\ \cap Ret_1|} \geq 0$$

Therefore, query expansion always guarantees that recall for the ontology-based model will be higher or equal to recall of keyword-based techniques. Note that if expanded concepts retrieve nothing or C_1 is itself a leaf concept, then the recall result will be the same in either case.

Similarly, using Eqs. 5 and 7, we get

$$\frac{P_o}{P_k} = \frac{\frac{\sum_{i=1}^{n} |Rel \cap Ret_i|}{\sum_{i=1}^{n} |Ret_i|}}{\frac{|Rel \cap Ret_1|}{|Ret_1|}} \tag{10}$$

$$\frac{P_o}{P_k} = \left(1 + \frac{|Rel \cap Ret_2| + \ldots + |Rel \cap Ret_n|}{|Rel \cap Ret_1|}\right) \times \left(1 - \frac{|Ret_2| + \ldots + |Ret_n|}{|Ret_1| + \ldots + |Ret_n|}\right) \tag{11}$$

The first term in Eq. 11 is greater than or equal to 1, i.e.,

$$\left(1 + \frac{|Rel \cap Ret_2| + \ldots + |Rel \cap Ret_n|}{|Rel \cap Ret_1|}\right) \geq 1$$

On the other hand, the second term in Eq. 11 is less than or equal to 1, i.e.,

$$\left(1 - \frac{|Ret_2| + \ldots + |Ret_n|}{|Ret_1| + \ldots + |Ret_n|}\right) \leq 1$$

Therefore, it is not a trivial problem to say which case is better.

Assume the best case; each C_i returns only relevant objects, so $Rel \cap Ret_i = Ret_i$ and $|Rel \cap Ret_i| = |Ret_i|$ for ∀i. Then, using Eq. 11, we get

$$\begin{aligned}\frac{P_o}{P_k} &= \frac{|Rel \cap Ret_1\ | + |\ Rel \cap Ret_2\ | + \ldots + |Rel \cap Ret_n\ |}{|Ret_1\ | + |Ret_2\ |\ + \ldots + |Ret_n\ |} \times \frac{|\ Ret_1\ |}{|Rel \cap Ret_1\ |} \\ &= \frac{|Ret_1\ | + |\ Ret_2\ | + \ldots + |Ret_n\ |}{|Ret_1\ | + |Ret_2\ |\ + \ldots + |Ret_n\ |} \times \frac{|\ Ret_1\ |}{|Ret_1\ |} \\ &= 1 \times 1 \\ &= 1\end{aligned}$$

Therefore, maximally, $P_o = P_k$.

Assume the worst case; each $C_i (i > 1)$ returns only irrelevant concepts. Then $Rel \cap Ret_i = 0$ and $|Rel \cap Ret_i| = 0$ for ∀i > 1. Then using Eq. 11,

$$\begin{aligned}\frac{P_o}{P_k} &= \frac{|Rel \cap Ret_1\ | + |\ Rel \cap Ret_2\ | + \ldots + |Rel \cap Ret_n\ |}{|Ret_1\ | + |Ret_2\ |\ + \ldots + |Ret_n\ |} \times \frac{|\ Ret_1\ |}{|Rel \cap Ret_1\ |} \\ &= \frac{|Rel \cap Ret_1| + 0 + \ldots + 0}{|Ret_1\ | + |Ret_2\ |\ + \ldots + |Ret_n\ |} \times \frac{|\ Ret_1\ |}{|Rel \cap Ret_1\ |} \\ &= \frac{|Rel \cap Ret_1|}{|Ret_1\ | + |Ret_2\ |\ + \ldots + |Ret_n\ |} \times \frac{|\ Ret_1\ |}{|Rel \cap Ret_1\ |} \\ &= \frac{|\ Ret_1\ |}{|Ret_1\ | + |Ret_2\ |\ + \ldots + |Ret_n\ |} \ll 1\end{aligned}$$

Therefore, at best $P_0 = P_k$, at worst $P_0 \ll 1$.

Using Eq. 3, F scores for keyword-based and ontology-based models are as follows:

$$Fscore_k = \frac{2 \times P_k \times R_k}{P_k + R_k} \tag{12}$$

$$Fscore_o = \frac{2 \times P_o \times R_o}{P_o + R_o} \tag{13}$$

Therefore, from Eqs. 13 and 14 we get

$$\frac{Fscore_o}{Fscore_k} = \frac{P_o \times R_o}{P_o + R_o} \times \frac{P_k + R_k}{P_k \times R_k} = \frac{\frac{1}{P_k} + \frac{1}{R_k}}{\frac{1}{P_o} + \frac{1}{R_o}} \tag{14}$$

Using Eqs. 4, 5, 6, and 7, we get from Eq. 15

$$\frac{Fscore_o}{Fscore_k} = \frac{\frac{|Ret_1\ |}{|Rel \cap Ret_1\ |} + \frac{|Rel\ |}{|Rel \cap Ret_1\ |}}{\frac{\sum_{i=1}^{n} |Ret_i\ |}{\sum_{i=1}^{n} |Rel \cap Ret_i\ |} + \frac{|Rel\ |}{\sum_{i=1}^{n} |Rel \cap Ret_i\ |}} \tag{15}$$

$$= \frac{\frac{|Ret_1\ | + |Rel\ |}{|Rel \cap Ret_1\ |}}{\frac{\sum_{i=1}^{n} |Ret_i\ | + |Rel\ |}{\sum_{i=1}^{n} |Rel \cap Ret_i\ |}} \tag{16}$$

$$\Rightarrow \frac{Fscoreo}{Fscorek} = \left(1 + \frac{|Rel \cap Ret_2| + \ldots + |Rel \cap Ret_n|}{|Rel \cap Ret_1|}\right) \times \frac{1}{1 + \frac{|Ret_2| + |Ret3| + \ldots + |Ret_n|}{|Ret_1| + |Rel|}} \quad (17)$$

The first term in Eq. 17 is greater than or equal to 1 (i.e., $1 + \frac{|Rel \cap Ret_2| + \ldots + |Rel \cap Ret_n|}{|Rel \cap Ret_1|} \geq 1$) and the second term in Eq. 17 is less than or equal to 1 (i.e., $\frac{1}{1 + \frac{|Ret_2| + |Ret_3| + \ldots + |Ret_n|}{|Ret_1| + |Rel|}} \leq 1$).

Therefore, we cannot say in a straightforward manner that one outperforms the other between $Fscore_k$ and $Fscore_o$. However, in special cases we can say which is better such as:

Again assume the best case; each C_i returns only relevant objects, so $Rel \cap Ret_i = Ret_i$ and $|Rel \cap Ret_i| = |Ret_i|$ for ∀i. Then, using Eq. 17, we get

$$\frac{Fscore_o}{Fscore_k} = \left(1 + \frac{\sum_{i=2}^{i=n} |Rel \cap Ret_i|}{|Rel \cap Ret_1|}\right) \times \frac{1}{1 + \frac{\sum_{i=2}^{i=n} |Ret_i|}{|Ret_1| + |Rel|}}$$

$$= \left(1 + \frac{\sum_{i=2}^{i=n} |Ret_i|}{|Ret_1|}\right) \times \frac{1}{1 + \frac{\sum_{i=2}^{i=n} |Ret_i|}{|Ret_1| + |Rel|}}$$

$$= \frac{1 + \frac{X}{A}}{1 + \frac{X}{A + |Rel|}}$$

$$\text{where } X = \sum_{i=2}^{i=n} |Ret_i| \text{ and } A = |Ret_1| \quad (18)$$

And it is obvious that

$$\frac{X}{A} > \frac{X}{A + |Rel|}$$

$$\Rightarrow 1 + \frac{X}{A} > 1 + \frac{X}{A + |Rel|}$$

$$\Rightarrow \frac{1 + \frac{X}{A}}{1 + \frac{X}{A + |Rel|}}$$

Then, using Eq. 18, we get

$$\frac{Fscore_o}{Fscore_k} > 1 \Rightarrow Fscore_o > Fscore_k$$

Assume the worst case; each $C_i (i > 1)$ returns only irrelevant concepts. Then $Rel \cap Ret_i = 0$ and $|Rel \cap Ret_i| = 0$ for ∀i > 1. Then, using Eq. 17, we get

$$\frac{Fscore_o}{Fscore_k} = \left(1 + \frac{\sum_{i=2}^{i=n} |Rel \cap Ret_i|}{|Rel \cap Ret_1|}\right) \times \frac{1}{1 + \frac{\sum_{i=2}^{i=n} |Ret_i|}{|Ret_1| + |Rel|}}$$

$$= \left(1 + \frac{0}{|Rel \cap Ret_1|}\right) \times \frac{1}{1 + \frac{\sum_{i=2}^{i=n} |Ret_i|}{|Ret_1| + |Rel|}}$$

$$= \frac{1}{1 + \frac{\sum_{i=2}^{i=n} |Ret_i|}{|Ret_1| + |Rel|}} < 1$$

$$\frac{Fscore_o}{Fscore_k} < 1 \Rightarrow Fscore_o < Fscore_k$$

The problem occurs when all or most C_i are irrelevant. However, by construction, the ontology's concepts are generally/almost always relevant especially in a small domain. We do not expect the worst case will happen in practice. However, its equivalent in the nonontology information retrieval (IR) research query expansion often produces irrelevant expansion terms.

7.1.2 Empirical results

We have developed a prototype system that was available on the Web (URL: http://esfahaan.usc.edu: 8080/examples/jsp/pac/pac3.jsp). A set of queries from real users were gathered through this interface they posed.

Sample queries are classified into three categories. The first category is related to broad/general query formulation such as "tell me about basketball", which is associated with an upper-level concept of the ontology. The second category is related to narrow query formulation such as "tell me about Los Angeles Lakers", which is associated with a lower-level concept of the ontology. The third category is context query, in which a user specifies a certain context in order to make the query unambiguous, such as "Laker's Kobe", "Boxer Mike Tyson", and "Team Lakers". For example, for keyword "Lakers" in our ontology, two concepts are selected: Major League Baseball player "Tim Laker" who plays for the team "Pittsburgh Pirates" and NBA team "Los Angeles Lakers". Although we collected data for all queries, results are reported for only nine. The reason for this is that the nine are chosen in a way that gives the worst result from an ontology-based model perspective.

The comparison metrics used for these two search techniques are precision, recall, and F score. First, we discuss precision, recall, and F score for individual queries. These queries are then grouped into the three categories: broad query, narrow query, and context query. Next, we present average precision, recall, and F score for each category and then for all the categories taken together.

In Figs. 5, 6, and 7, the x-axis represents sample queries. The first three queries are related to broad query formulation, the next three to narrow query formulation, and the last three to context queries. In Figs. 5, 6, and 7, for each query the first and second bars represent the recall/precision/F score for ontology and keyword-based search techniques, respectively. Corresponding numerical values are reported in Table 1. Although

Table 1. Recall/precision/F score for two search techniques

Types of Queries		Recall Ontology	Keyword	Precision Ontology	Keyword	F score Ontology	Keyword
Generic	Query 1	90%	11%	98%	95%	94%	20%
/broader	Query 2	95%	15%	89%	90%	92%	26%
queries	Query 3	87%	30%	100%	82%	93%	44%
Specific	Query 4	90%	76%	76%	90%	83%	83%
/narrow	Query 5	85%	71%	100%	72%	91%	71%
queries	Query 6	100%	65%	77%	100%	87%	79%
Context	Query 7	90%	76%	76%	29%	82%	42%
queries	Query 8	85%	83%	100%	34%	92%	49%
	Query 9	100%	74%	74%	16%	85%	27%
	Averages	91.3%	55.6%	87.7%	67.5%	88.7%	49%

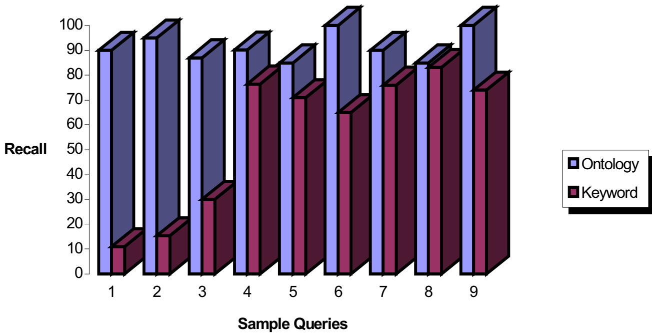

Fig. 5. Recall of ontology-based and keyword-based search techniques

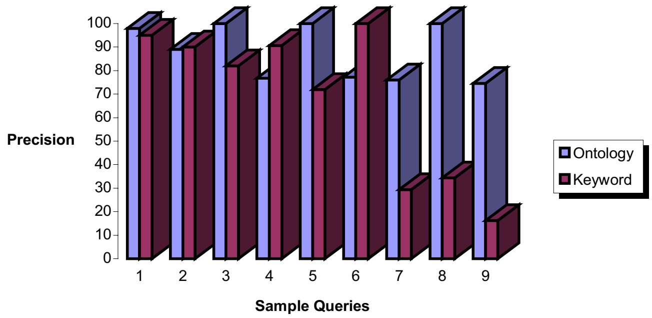

Fig. 6. Precision of ontology-based and keyword-based search techniques

the vector space model is rank-based and our ontology-based model is a Boolean retrieval model, in the former case we report precision for maximum recall in order to make a fair comparison.

In Fig. 5, the data demonstrate that recall for our ontology-based model outperforms recall for keyword-based techniques. Note that this pattern is pronounced in relation to broader query cases. For example, in query 1, 90% vs. 11% recall is achieved for ontology-based as opposed to keyword-based techniques, whereas for query 4, 90% and 76% recall are obtained. This is because in the case of a broader query, more child concepts are added, as compared to narrow query formulation or a context query case. Furthermore, in a context query case, it is usual for broader query terms to give context only. In an ontology-based model, these terms will not participate in the query expansion mechanism. Instead, broader query terms will be subsumed under specific concepts. For example, in query 7, the user requests “tell me about Team Lakers”. Concepts referring to “team” will not be expanded. Therefore, the gap between the two techniques is not pronounced.

In Fig. 6, for broader query cases, usually the precision of the ontology-based model outperforms the precision of the keyword-based technique. This is because our disambiguation algorithm disambiguates upper-level concepts with greater accuracy compared to lower-level concepts. For example, the disambiguation algorithm for metadata acquisition chooses the most appropriate region for each audio object. Recall that a region is formed by a league, its team, and its players. Thus if a query is requested in terms of a particular league, that is, in relation to an upper concept in this region, precision will not be hurt. However, the algorithm might fail to disambiguate lower-level concepts in that region (e.g., players). For a narrow query formulation case, the precision obtained in the ontology-based model may not be greater than that obtained through use of the keyword-based technique. In query 4, the user requests “tell me about Los Angeles Lakers”. In the ontology-based model, the query is expanded to include all this team’s players. It might be possible during disambiguation in metadata acquisition for some of these players to be associated with audio objects as irrelevant concepts, in particular when disambiguation fails. Some relevant concepts, such as other players, are also associated with these audio objects. Thus for our ontology-based model these objects will be retrieved as a result of query expansion, leading to a deterioration in precision. In the keyword-based case, we have not expanded “Lakers” in terms of all of the players on the Lakers team. Therefore, we just look for the keyword “Lakers” and the abovementioned irrelevant objects associated with its group of players will not be retrieved. Thus in this instance, we observed 76% and 90% precision for ontology-based and keyword-based techniques, respectively.

In the case of the context query, it is evident that the precision of the ontology-based model is much greater than that of the keyword-based model. Since in the ontology-based model some concepts subsume other concepts, audio objects will only be retrieved for specific concepts. On the other hand, a search using keyword-based techniques looks for all keywords. If the user requests “team Lakers”, the keyword-based

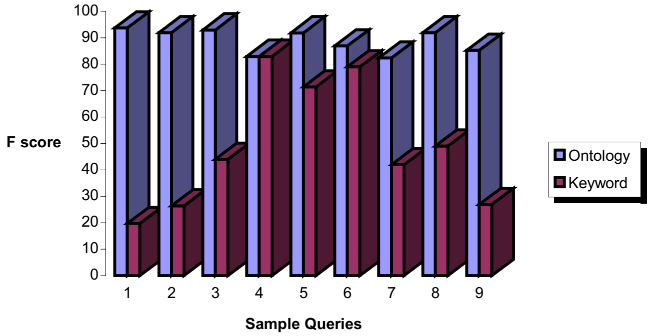

Fig. 7. F score of ontology-based and keyword-based search techniques

technique retrieves objects with the highest rank when the keywords "team" and "Lakers" are present. Furthermore, in order to facilitate maximum recall, we have observed that relevant objects will be displaced along with irrelevant objects in this rank. Note that some irrelevant objects will also be retrieved that only contain the keyword "team". Thus, for query 7, levels of precision of 76% and 29% have been achieved.

Finally, the F score of our ontology-based model outperforms (or at least equals) that of a keyword-based technique (see Fig. 7). For the broader and context query case, precision and recall are usually high for the ontology-based model in comparison with the keyword-based technique. Therefore, F score differences for the ontology-based model are also pronounced. For example, for query 1, the F scores for ontology-based and keyword-based techniques are 94% and 20%, respectively. For the narrow query case, the F score of our ontology-based model is slightly better than or equal to that of the keyword-based technique. For example, in query 4, we observed a similar F score (83%) in both cases; however, in queries 5 and 6, we observed that the F score of the ontology-based model (91%, 87%) outperformed the keyword-based technique, (71%, 79%).

8 Conclusions and future work

In this paper, we have proposed a potentially powerful and novel approach for the retrieval of audio information. The crux of our innovation is the development of an ontology-based model for the generation of metadata for audio and the selection of audio information in a user-customized manner. We have shown how the ontology we propose can be used to generate information selection requests in database queries. We have used a domain of sports news information for a demonstration project, but our results can be generalized to fit many additional important content domains including but not limited to all audio news media. Our ontology-based model demonstrates its power over keyword-based search techniques by providing many different levels of abstraction in a flexible manner with greater accuracy in terms of precision, recall, and F score.

Although we are confident that the fundamental conceptual framework for this project is sound and its implementation completely feasible from a technical standpoint, the most pressing question remaining relates to the cost of building such domain-specific ontologies and connecting domain data to them automatically. Ongoing ontology construction research in the knowledge representation community is addressing these questions [29,2]. In addition, other questions remain to be answered in future work. These include detailed work on evolving ontologies, extracting highlighted sections of audio, dynamic updates of user profiles, addressing retrieval questions in the video domain, and facilitation of cross-media indexing.

We would like to build ontology that is easy to update, open and dynamic both algorithmically and structurally for easy construction and modification, and fully capable of adapting to changes and new developments in a domain. For example, suppose player "Bryant Kobe" switches from team "Los Angeles Lakers" to team "Portland Trail Blazers". In this case, we need to remove the interrelationship link between concepts "Bryant Kobe" and "Los Angeles Lakers" and add a new link between the concepts "Bryant Kobe" and "Portland Trail Blazers". In this connection, we would like to address the problem of how to create useful ontology by minimizing the cost of initial creation while allowing for novel concepts to be added with minimum intervention and delay. For this, we would like to combine techniques from knowledge representation [19], resource description framework [26], natural language processing, and machine learning.

Users may be interested in highlights of the news. For this, we need to identify and store these highlights. By analyzing the pitch of the recorded news, we can identify sections to be highlighted [12]. This is because, as is well known in the speech and linguistics communities, there are changes in pitch under different speaking conditions. For example, when a speaker introduces a new topic, the range of pitch will be increased. On the other hand, subtopics and parenthetical comments are often associated with a compression of pitch range. Since pitch varies considerably between speakers, it is also necessary to find an appropriate threshold for a particular speaker.

We would like also to address the problem of customization through assessing ways to dynamically update user profiles. We are confident that we will ultimately be able to develop an intelligent agent that will dynamically update user profiles. This will provide a level of customization that can have broad application to many areas of content and user interest.

References

1. Adali S, Candan KS, Chen S, Erol K, Subrahmanian VS (1996) Advanced video information system: data structures and query processing. ACM Multimedia Sys J 4:172–186
2. Agirre E, Ansa O, Hovy E, Martinez D (2000) Enriching very large ontologies using the WWW. In: Proceedings of the ontology learning workshop, ECAI, Berlin, Germany, August 2000
3. Arons B (1993) SpeechSkimmer: interactively skimming recorded speech. In: Proceedings of the ACM symposium on user interface software and technology, Atlanta, November 1993, pp 187–196
4. Aslan G, McLeod D (1999) Semantic heterogeneity resolution in federated database by metadata implantation and stepwise evolution. VLDB J 18(2):120–132
5. Baeza R, Neto B (1999) Modern information retrieval. ACM Press, New York, Addison-Wesley, Reading, MA

6. Bunge M (1977) Treatise on basic philosophy. Ontology I. The furniture of the world, vol 3. Reidel, Boston
7. Gibbs S, Breitender C, Tsichritzis D (1994) Data modeling of time based media. In: Proceedings of ACM SIGMOD international conference on management of data, Minneapolis, June 1994, pp 91–102
8. Gonzalo J, Verdejo F, Chugur I, Cigarran J (1998) Indexing with WordNet synsets can improve text retrieval. In: Proceedings of the Coling-ACL'98 workshop: usage of WordNet in natural language processing systems, Montreal, August 1998, pp 38–44
9. Gruber TR (1993) A translation approach to portable ontology specifications. Knowl Acquisition 5(2):199–220
10. Guarino N, Masolo C, Vetere G (1999) OntoSeek: content-based access to the Web. IEEE Intell Sys 14(3):70–80
11. Hauptmann AG (1995) Speech recognition in the informedia digital video library: uses and limitations. In: Proceedings of the 7th IEEE international conference on tools with AI, Washington, DC, November 1995
12. Hirschberg J, Grosz B (1991) Intonational features of local and global discourse. In: Proceedings of the speech and natural language workshop, Harriman, NY, February 1991, pp 23–26
13. Hjelsvold R, Midstraum R (1994) Modeling and querying video data. In: Proceedings of the 20th international conference on very large databases (VLDB'94), Santiago, Chile, September 1994, pp 686–694
14. Khan L, McLeod D (2000) Audio structuring and personalized retrieval using ontologies. In: Proceedings of IEEE advances in digital libraries, library of congress, Bethesda, MD, May 2000, pp 116–126
15. Khan L, McLeod D (2000) Effective retrieval of audio information from annotated text using ontologies. In: Proceedings of the ACM SIGKDD workshop on multimedia data mining, Boston, August 2000, pp 37–45
16. Labrou Y, Finin T (1999) Yahoo! as an ontology – using Yahoo! categories to describe documents. In: Proceedings of the 8th international conference on knowledge and information management (CIKM-99), Kansas City, MO, October 1999, pp 180–187
17. Lenat DB (1995) Cyc: a large-scale investment in knowledge infrastructure. Commun ACM 38(11):33–38
18. Miller G (1995) WordNet: a lexical database for English. Commun ACM 38(11):39–41
19. Mitra P, Kersten M, Wiederhold G (2000) Graph-oriented model for articulation of ontology interdependencies. In: Proceedings of the 7th international conference on extending database technology (EDBT 2000), Konstanz, Germany, March 2000, pp 86–100
20. Omoto E, Tanaka K (1993) OVID: design and implementation of a video-object database system. IEEE Trans Knowl Data Eng 5(4):629–643
21. Peat HJ, Willett P (1991) The limitations of term co-occurrence data for query expansion in document retrieval systems. J ASIS 42(5):378–383
22. Rabiner LR, Schafer RW (1978) Digital processing of speech signals. Prentice-Hall, Upper Saddle River, NJ
23. Salton G (1989) Automatic text processing. Addison-Wesley, Reading, MA
24. Smeaton AF, Rijsbergen V (1993) The retrieval effects of query expansion on a feedback document retrieval system. Comput J 26(3):239–246
25. Swartout B, Patil R, Knight K, Ross T (1996) Toward distributed use of large-scale ontologies. In: Proceedings of the 10th workshop on knowledge acquisition for knowledge-based systems, Banff, Canada, 1996
26. Using XML: Ontology and Conceptual Knowledge Markup Languages (1999) http://www.oasis-open.org/cover/xml.html
27. Voorhees E (1994) Query expansion using lexical-semantic relations. In: Proceedings of the 17th annual international ACM SIGIR conference on research and development in information retrieval, Dublin, Ireland, July 1994, pp 61–69
28. Wilcox LD, Bush MA (1992) Training and search algorithms for an interactive wordspotting system. In: Proceedings of the IEEE conference on acoustics, speech, and signal processing, San Francisco, vol 2, pp 97–100
29. Woods W (1999) Conceptual indexing: a better way to organize knowledge. Technical report of Sun Microsystems