An Information Retrieval Application using Ontologies

Christian Paz-Trillo & Renata Wassermann
Departament of Computer Science – Institute of Mathematics and Statistics
University of São Paulo, Brazil
{cpaz,renata }@ime.usp.br

Paula P. Braga
Department of Philosophy
University of São Paulo, Brazil
pbraga@usp.br
October 13, 2005

Abstract

Searching for information in long videos can be a time-consuming experience. In this paper, we describe OnAIR, an ontology-aided information retrieval system applied to retrieve clips from video collections.

We used a video collection compiled from interviews with Ana Teixeira, a Brazilian artist. The interviews were made by Paula P. Braga, the domain expert. The interview is developed in the domain of contemporary art and the system uses a domain ontology to expand the queries with related terms. We tested the system with a battery of queries, and we veri.ed that the ontology contributes to the e.ciency improvement in terms of the relevance of retrieved documents. We designed the system to work in a domain-independent way, allowing us to move to other domains by just changing the underlying ontologies and video collections.

Keywords: Ontologies, Information Retrieval, Video Retrieval, Query Expansion.

1. Introduction

In this paper, we describe OnAIR (Ontology-Aided Information Retrieval), an ontology-based video retrieval system, that can be used to allow users to look for information in video fragments through queries in natural language. The idea is to save the user from the time consuming experience of having to browse through hours of video in order to find an answer for his questions.

We tested this application in a set of recorded interviews with Ana Teixeira, a Brazilian artist whose work deals mainly with urban interventions. In this domain, it allows for a new relationship between the art spectator and the work of art: the visitor of a museum or art gallery will be able to develop his/her own questions and thoughts about the exhibited works. Exhibition spaces usually offer only wall or catalogue texts for the visitors, frequently written in a very specialized language, and focusing on the curatorial thesis, which leaves no room for the spectator's creative thought. In some cases, such as contemporary art works that privilege the participation of the spectator on the making of the work, such museological practices represent an unacceptable gap and an inconvenient mediation between the goals of the artist and the experience of the museum visitor.

Ana Teixeira develops her work on the streets or other public spaces, frequently involving passers by in the work. Bringing her work to a conventional exhibition space asks for new approaches to the relationship between the visitor and the museum.

OnAIR offers an interaction mechanism between the spectator and the artist, which is helpful in the context

of contemporary art. The interviews with Ana Teixeira, that were recorded in digital video, served as a guide to the development of a contemporary art ontology, and we expect to have a refinement process as we record interviews with other contemporary artists. OnAIR uses ontologies to improve the relevance of retrieved video fragments.

One of the main features of OnAIR is to be a flexible system, that can be configured to different kind of videos (courses, seminars, etc.) on different domains by extending or changing underlying ontologies and organizing a clip database for it. Section 2 introduces the Information Retrieval problem and approaches to solve it that will be used in this project. The Ontology developed is described in Section 3 as well as the tools being used to create and access it from the system. In Section 4 we describe OnAIR architecture, detailing its two main processes: indexing and retrieval. Section 5 describes the results we obtained when testing the system in a contemporary art domain. In Section 6 we present related works on video retrieval, ontology-based information retrieval and question answering systems. Finally, in Section 7, we present our concluding thoughts and discuss some ongoing and future work.

2. Information Retrieval

2.1 Definition

An Information Retrieval (IR) System allows users to look for information in a collection of documents (or other information sources) through queries usually formatted as a set of keywords[2]. Using the query, an IR system retrieves information that might be relevant to the user.

2.2 IR problems

Keyword-based IR systems are limited in its ability to distinguish between relevant and irrelevant texts [17], mainly due to:

- **Synonymy:** This is the case when there are several terms to describe the same object (or concept). A keyword-based IR system will only retrieve those documents that refer to the object (or concept) by the same term used in the query. For instance, a query to look for information about accommodations in Rio de Janeiro could be "Inn Rio de Janeiro", and only documents containing the word "Inn" would be retrieved. The system would miss information associated with words such as "hotel" or "hostel".
- **Polysemy:** A polysemic word is a word with multiple meanings, i.e., expressing different concepts.

In this case a query might lead to the retrieval of documents which deal with concepts foreign to the subject of interest. In the contemporary art domain, the term "Reception" might refer to a party (for example, an exhibition opening) or to the impact of a work of art on the spectator. Thus, "reception" might be interpreted in the first sense when it is related to a place, while it probably refers to the second meaning when it is associated to a person.

We refer to these limitations as the *keyword barrier* [17]. To go beyond this obstacle, an IR system needs another structure for the data with which it works, e.g., a structure based on concepts instead of mere keywords. In our system, the concepts are organized in a relational and hierarchical structure, an ontology, that solves the synonymy problem. Polysemy is a very complex problem and, trying to solve it usually involves semantic analysis of the text being processed. However, when the domain is restricted, less cases of ambiguity arise. In abstract domains like contemporary art some examples are present, like "Reception". We expect that dealing with more technical domains, like Object-Oriented Programming (See Section 7), leads to minimize polysemy.

2.3 Performance Measures for IR systems

There are standard measures to evaluate the performance of IR systems [17]:

- **Precision:** The ratio of documents retrieved by the system that are actually relevant to the query divided by the total number of documents retrieved.

$$P = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of documents retrieved}} \quad (1)$$

For instance, if the system retrieved 6 documents for a query, where 3 of them were actually relevant, the precision performance for the system in that query is 0.5 or 50%. Polysemy may produce low precision rates, because irrelevant documents might be retrieved.

- **Recall:** There may be many documents in the database that the user considers relevant, but only some of them will be retrieved by the system. The recall performance of a query is the number of relevant documents retrieved by the system divided by the total number of relevant documents in the database.

$$R = \frac{\text{Number of relevant documents retrieved}}{\text{Total of relevant documents in the collection}} \quad (2)$$

Recall is difficult to measure, because it requires knowledge about the total number of relevant documents in the database, which has to be determined manually. To measure recall, an upper bound is usually established by finding documents that should have been retrieved but were not. The greater the manual effort to find these documents, the more precise the recall measure is. Synonymy leads to lower recall rates, because relevant documents referring only to synonyms of a word used in the query may not be actually retrieved.

- **Response time:** The elapsed time between the submission of a query and the presentation of the documents retrieved by the system.

Precision could be easily maximized by retrieving a single document that is certainly relevant, and recall by retrieving all documents in the database. Thus, a measure that combines both of them is preferred, for example, the *F-measure* [26]

$$F = 2\frac{R\ P}{R+P}$$

where F-measure is the harmonic mean of precision and recall. The advantage of using F-measure is that maximizing it means maximizing a combination of recall and precision.

3. Ontologies in IR

Recently, ontologies have been used in Information Retrieval to improve recall and precision [1, 9, 13, 11]. Its principal use is related to query expansion, which consists in looking for the terms in the ontology more related to the query terms, to use them as a part of the query.

We developed an ontology that contains the concepts used in the interview with Ana Teixeira. The concepts refer mainly to people, institutions and objects related to contemporary Brazilian art. We also included concepts related to more general ideas like feelings, countries or places, where ontology reuse could have helped our work. However, because the system works with Brazilian Portuguese and most shared ontologies are developed in English, little ontology reuse was possible. In the future, we intend to turn it into a bilingual ontology supporting both English and Brazilian Portuguese.

In this section we explore the ontology development process, how we used ontologies to improve precision and recall and which ontology tools supported our development.

3.1 Development Process

The exercise of developing an ontology about contemporary art raised several philosophical thoughts we consider important to mention here. Today art deals with the effacing of frontiers regarding classifications. It is impossible to constrict a work of art to the domain of a category like "painting" or "sculpture". New classifications appear from time to time to work around this language limitation: one may classify a work of art as "performance" because the artist paints a canvas while dancing around it during the exhibition, or "installation" because the spectator is confronted with an environment that includes music, video, sculptures, smells, all making a single work of art[1].

During the development of the contemporary art ontology, we dealt with both the awareness that contemporary art defies categories and the necessity to develop a machine-oriented, class-based model of the art universe. Thus, limiting the world to a machine understandable system of concepts and categories, we started modeling the art world with jargon terms such as "institution", "work of art", "place", "text", "individual", and dismembering each of these main blocks into sub-classes in an "is-a" relationship. Therefore, an "institution" could be a museum, a gallery, a cultural center, a company, a school, and so on. Later, we examined all the keywords we extracted from the interviews with the artist Ana Teixeira, detecting the necessity to include additional super-classes such as "Material" and "Power". We tried to have classes to encompass all the words she mentioned in her talk that the domain expert considered relevant. The next step was to associate a list of synonyms to each word that appeared in the class tree[2]. Later, we developed relationships between classes to accommodate terms that the class system missed and we are currently registering instances into our model.

This procedure will be repeated as we interview other contemporary artists, detailing more and more the ontology. The flexibility of the ontology system allows us to reevaluate our model from time to time, a necessary step in any system of thought. For details about the current state of our ontology, see http: //www.ime.usp.br/~cpaz/ontologies/arte/.

In Figure 1 we show an excerpt of the ontology developed. It includes the two base classes Institution

[1]See for example the exhibition catalog "Blurring the Boundaries: Installation Art 1969-1996". Museum of Contemporary Art, San Diego, 1997.

[2]Synonyms are represented as comments of the class, to facilitate maintenance, but after an automatic processing they will be converted to a language construction that establishes an equivalence relationship between two classes.

and EVENT, these two classes are related by an ORGANIZES relationship, meaning that an institution organizes events, and events are organized by institutions.

Sub-classes of Institution includes FOUNDATION and GALLERY. STORE is considered a synonym of Gallery in this context, and it is shown as a two-way "is-a" relationship.

Instances are also shown, for example: VIZINHOS is an EXHIBITION, and was organized by the Gallery VERMELHO. BIENAL FOUNDATION organizes the BIENAL OF SÃO PAULO event.

3.2 Ontology Tools

We are currently using OWL [24](Web Ontology Language) to represent the ontology, and it is theW3C recommendation for describing ontologies, derived from the Resource Definition Framework (RDF)[7].

The specification of OWL is divided in three sub languages: OWL Full, with high expressivity but there is no guarantee of computability for all of its features; OWL DL based on description logics, offers the maximum expressiveness that guarantees completeness and decidability and OWL Lite that is less expressive, but has the lowest computational complexity.

In the current implementation we only use OWL Lite features, but we intend to expand the expressivity used, so using OWL will help us to easily add extra features with minor changes in implementation. Additionally, even if in the present application we were not able to reuse other ontologies, as OWL is becoming a standard, the possibility of reutilization is higher than with other ontology languages.

We used Protégé [8] to elaborate the ontology, because it offers an easy-to-use interface and its OWL-plugin [15] allows exporting ontologies into the OWL language.

To explore the ontology from inside the system, we used Jena [3][18], an open source Java API that has a comprehensive subsystem to manipulate ontologies, and that allows to explore OWL.

4. OnAIR Architecture

OnAIR is in essence an information retrieval system [26, 2]. And, as such, it has two main processes: indexing and retrieval. The indexing process takes a collection of documents and other information (a domain ontology and keywords associated to each document) and generates the structures needed to allow the retrieval process to use it to respond to user queries.

In this section, we describe both processes.

4.1 Indexing Process

The indexing process is responsible for the creation of the structures that the retrieval process will use.

The input of this process consists in:

- **A Video Collection**: In the case of OnAIR, the collection consists in a set of short clips, manually extracted from a long interview or lecture. Each clip is associated with: (1) a set of manually assigned keywords, selected by the domain specialist; (2) a set of resources (images) that can be shown during the video in a specified period and, optionally; (3) a transcription of the speech.
- **A Domain Ontology**: An ontology as described in Section 3 in OWL format. It is used in the retrieval process to expand the original user queries (Section 4.2).

The indexing process is executed through an administrative application that allows a system administrator to register the video clips, the ontology and establish other configuration values.

This process produces the three following components:

- **XML Configuration File**: Contains the associations between the clips and their transcriptions, keywords and resources, and some general configuration data.
- **Inverted Index**: An inverted index is a structure that stores the frequency and the occurrences of the terms in a collection of documents [12]. It also stores the weight of terms in a document, to avoid computing it in the retrieval phase. The terms are saved after their affixes are removed by a stemming process. In OnAIR's first application we used the RSLP[4] algorithm proposed by [20]. Stopwords, i.e., common words such as articles or prepositions, are not included in this index because they do not help to determine the relevance of the document to a query.

[3] http://jena.sourceforge.net/

[4] RSLP: Portuguese Language Su.x Remover

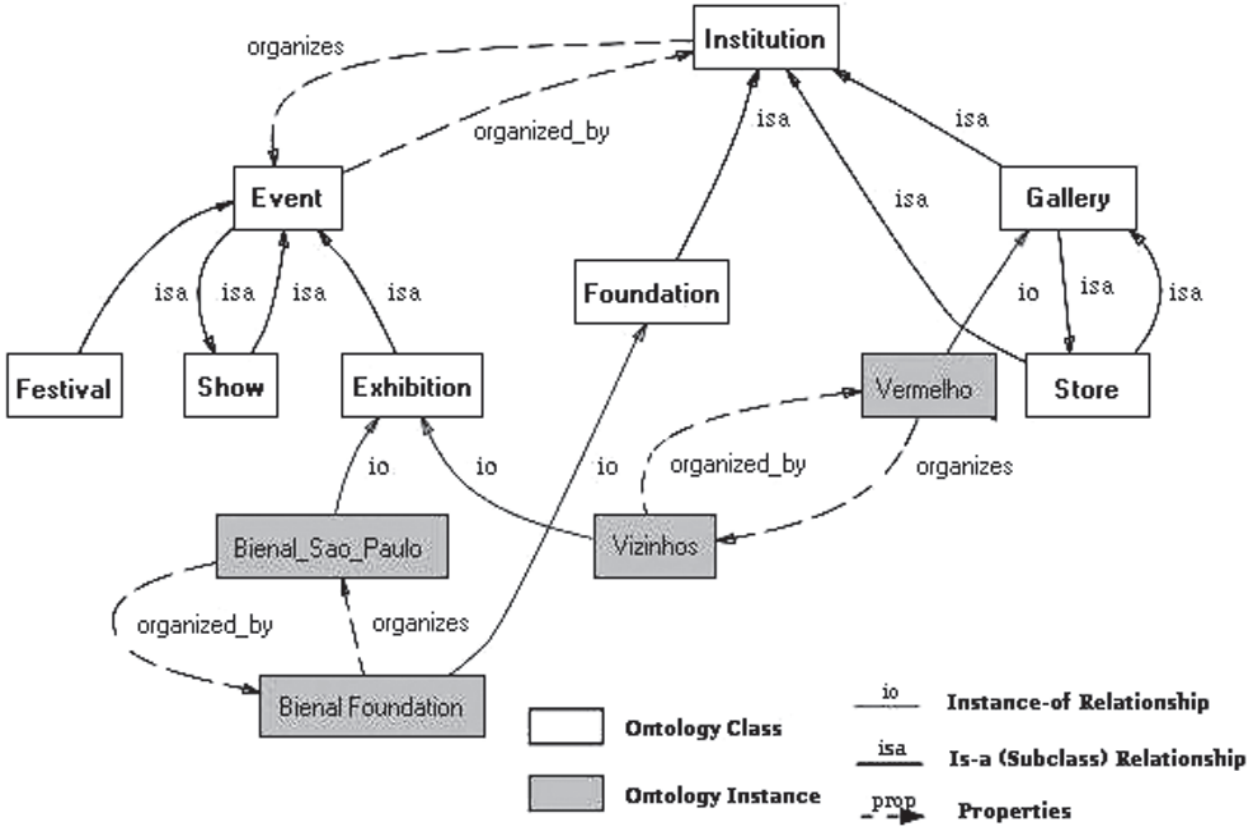

Figure 1: Ontology excerpt.

An index using the text in video transcriptions and a second one using just the keywords registered by the domain expert can be generated, depending on the information availability.

The algorithm that generates the inverted index from full text is shown in Figure 2. It goes through the documents getting the full text of each one and counting the occurrence of terms in the collection. When a token is a compound word in the ontology it is specially stemmed, considering the stemming of each word of the compound word. Finally it computes the weights of words in documents using the TF-IDF[5] weighting scheme [23].

```
1  Algorithm CreateIndex(collection, stemmer, onto) {
2
3     For Each Document doc in collection {
4        doc_entry = index.addDocEntry(doc.id);
5        For Each Token tok in doc.fullText {
6           If tok is compound word tok_ont in onto {
7              tok = stemmer.stem(tok_ont);
8           } Else {
9              tok = stemmer.stem(tok);
10          }
11          doc_entry.countOccurrence(doc, tok);
12          If doc_entry is not in index.termEntries {
13             index.addTermEntry(tok);
14          }
15          term_entry = index.getTermEntry(tok);
16          term_entry.countOccurrence(doc, tok);
17       }
18    }
19    index.computeWeightTerms():
20    Returns index;
21
22 }
```

Figure 2: Algorithm for Inverted Index generation.

• **Ontology Data Structure**: OWL is a very expressive language, but in this application we did not use all of its representational power. Given that Ontology engines, like Jena, are prepared to deal with much of OWL expressiveness, we designed a simplified data structure that simulates the ontology behavior used by the retrieval process. This allowed us to improve the performance of the system in terms of processing time[6]. Subclass relationships, instances, equivalent classes and properties can be modelled as a graph, and we actually implement a little inference engine over this graph.

The indexing process is computationally expensive but it needs to be executed only when the video collection or the ontology are modified. It is an off-line processing that does not affect the response time to queries.

4.2 Retrieval Process

Once the indexing process is executed, the configuration .le, the inverted index and the ontology data structure are used by the retrieval process, implemented by a visualizer (See Figure 3 for a screenshot).

This application allows the user to enter a query in natural language and shows to the user a list containing the videos that better answer the query ordered by relevance. It also allows the user to watch sequentially the listed videos or manually select one of them.

The query captured in this figure was: *"how do you sell your work?"* and we will use it as an example during the retrieval process description. The retrieval process is shown in Figure 4 and its subprocesses are described here:

- **Pre-processor**: In this subprocess, a misspelling detection is applied to the user query. The Jazzy API[7] is used with a general dictionary (we used a Brazilian Portuguese dictionary, br.ispell [25]) and a domain dictionary automatically extracted from the domain ontology, during the indexing process. Suggestions are presented with the results of the query, so the user can manually reformulate his/her query.

  Besides the misspelling detection, stopwords are disconsidered, and the stemming process is applied to the query. Finally, weights are assigned to the terms in the query, based on their presence or absence in the ontology and their frequency in the collection. For example, in the query *"How do you sell your work?"*, *"how"*, *"do" "you"* and *"your"* are stopwords, *"work"* receives a eight of 1, and *"sell"* receives a weight of 2 because besides being in the ontology it has low frequency in the collection.

- **Query Expansion**: For each term in the collection, its similarity to the query is computed. This similarity is a weighted average of the similarity between the term and each query term. A synonym of a term, expressed as an equivalent class in OWL, has the maximum similarity value: 1.

[5]Term Frequency - Inverse Document Frequency (TF-IDF) gives a measure of the importance of a term within a document. It is directly proportional to the frequency of the term in the document and inversely proportional to the number of documents in which the term appears.

[6] Despite of this fact, we kept the implementation using Jena, because ontology engines are in constant development and possibly, in future versions, its performance will become competitive.

[7]Jazzy is an open-source Java API for misspelling correction and it is available at http://jazzy.sourceforge.net/

Figure 3: OnAIR visualizer application.

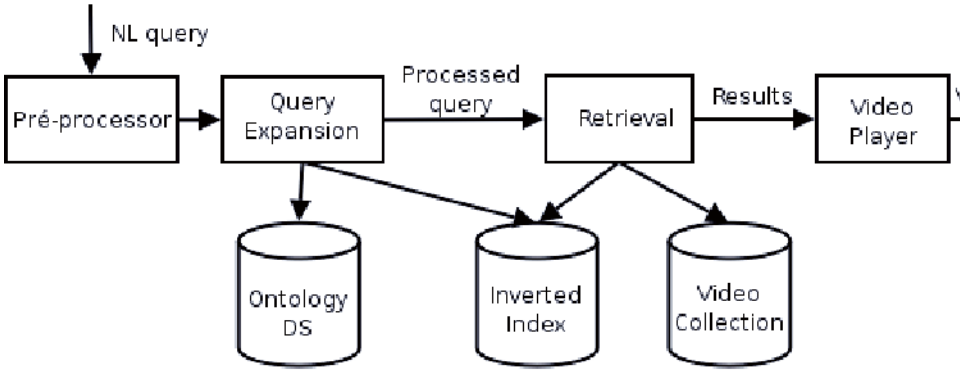

Figure 4: Retrieval process.

We used an approach that combines the use of the class hierarchy in the ontology, the terms frequency in the collection and the relationships in the ontology to compute the similarity. The algorithm for query expansion is shown in Figure 5. In this algorithm the similarity of each term in the index with the query is computed, represented by the weighted average of the similarity with all query terms. The similarity between two terms in the ontology is computed via the algorithm shown in Figure 6. This computation is derived from the schema proposed in [16] considering additionally a term that weights the properties linking both terms with respect to the total of properties they have.

The *r* most similar terms to the query are added to the original query, where *r* is a system parameter.

In the former example, the expansion mechanism would add the terms *"store"* and *"gallery"* to the query because they are very related to *"sell"* in the ontology.

- **Retrieval**: In this stage, the query is compared with the video clips contents (transcriptions or keywords) and the system retrieves those which best answer the query. We used the vector space model [23, 12] for retrieval. Both indexes generated during the indexing process can be used in conjunction and combined by a factor specified in the XML configuration file. In the example, the clip *"Process - Orders"* contains, in the keyword-based index, the terms: *"store"* and *"sell"*, so it has the highest similarity to the query and it is listed first by the retrieval process. Other clips listed just contain one of the terms related to *"sell"*.
- **Video Player**: The system presents the list of videos ordered by relevance to the query. The user can then select which videos to play or watch all of them in sequence. We implemented the video player using the Java Media Framework[8], which offers basic video functionalities such as play, stop, and pause.

5. Results

In this section we describe the results of a battery of tests executed on OnAIR. Tests were executed over five configurations, shown in table 1. The first is the system with no query expansion using keyword index.

For each of the ontology engines used (Jena and the data structure), there are two configurations, one using just the keyword index and another using both indexes.

[8] http://java.sun.com/products/java-media/jmf/

Mnemotecnic	Engine	$w_{keyword}$	$w_{transcr}$
noexpand	-	1,0	0,0
ds-keyw	OntDS	1,0	0,0
jena-keyw	Jena	1,0	0,0
ds-both-05	OntDS	0,5	0,5
jena-both-05	Jena	0,5	0,5

Table 1: Configurations for the tests.

We used a set of fifty queries elaborated by the domain expert from the perspective of the possible user of the system, a visitor of the museum in an art exposition of Ana Teixeira. These tests led us to two analysis: a global one, to evaluate the global behavior of the system and, a local one, with two selected queries to evaluate the system behavior in more detail under specific conditions.

5.1 Global Tests

We executed the fifty queries in the five configurations with a fixed maximum of documents to retrieve and fixed minimum relevance acceptance value of 5 and 0.1 respectively. These values were determined experimentally. Figure 7 shows the average and confidence intervals of F-measure for each configuration. We computed the average F-measure for the fifty queries on each configuration, and its confidence interval was generated by its standard deviation.

As we can see, configurations using both indexes and query expansion performed better in average, while the configuration without expansion showed the lowest average. The variance in the results is significant mainly because the video collection used for test is yet small, and small variations in retrieved documents affect strongly precision and recall.

These global tests allowed to see the general behavior of the system. We made a more detailed analysis for two queries that presented relevant characteristics, and we considered just the configurations with no expansion and with expansion using the ontology data structure.

```
Algorithm ExpandQuery(collection, query, index, onto) {

   query_weight = 0;
   For Each Term u in query {
      query_weight += query(u.weight);
   }

   For Each Term t in index {
      For Each Term u in query {
         sim(u) += query_weight * TermsSim(t, u, onto, index);
      }
      wex(t).peso = sim(t) / query_weight;
   }

   For Each Term u in query {
      If wex(t) is one of max_exp terms with higher weight {
         query_weight += query(u.weight);
      }
   }

}
```

Figure 5: Algorithm for Query Expansion.

5.2 Local Tests

We selected two queries from the battery. The queries are written in Brazilian Portuguese as it is the language of the videos:

- *"Por que você não dá então alguma coisa útil ao povo que está passando necessidade?"* (*"Then why do not you give something useful to the people who are starving?"*).
- *"Como você vende sua obra?"* (*"How do you sell your work?"*).

For both queries we show two graphics: (a) n-point average precision and; (b) a f-measure vs. retrieved documents graphic that allows us to determine the optimal number of retrieved documents for each query.

The n-point average precision graphic is commonly used in information retrieval research. To obtain these graphics, the number of documents retrieved is incremented to get different degrees of recall (varying from 0 to 1, when possible) and the precision obtained for each of them is registered.

In the first query, as we can see in Figure 8, no expand did not retrieve any relevant document, we got low precision values when using both indexes and low recall values when using just the keyword index.

In Figure 8b we can see that the highest values of F-measure were obtained when more than fifteen documents were retrieved. We detected that the abundance of irrelevant words, like *"then"* and *"some"*, could be the factor that affected system performance. To validate this affirmation, we re-elaborated the query as *"Why do you not give something to people who needs"*.

This query has less irrelevant words, and as shown in Figure 9, it allowed OnAIR get the highest F-measure values faster, even though precision remained still low. In the second query, configurations with expansion performed better. We got high precision values and reached a recall of 100% when using both indexes. The best F-measure was obtained when OnAIR retrieved five documents. We can see this results in Figure 10.

This query uses words that are present in the ontology, and query expansion mechanism allows to retrieve the relevant fragments to the user. The word *"sell"* is related to *"gallery"*, *"store"* and *"market"*, and the videos relevant to the query are related to some of these related words.

We can conclude that using ontologies to expand the queries, in this domain, helps to increment the recall, and that it is necessary to guide the users to avoid using redundant words, maybe offering more interactivity.

We have also performed some tests in chich only the class hierarchies was used leaving all other properties out. The tests showed that the recall was 5% higher when the properties were used, allowing us to conclude that a rich ontology gives better results for retrieval than a simple taxonomy.

```
1
2  Algorithm TermsSim(t1, t2, onto, index) {
3
4     If t1 == t2 Returns 1.0;
5
6     cls1 = onto.getOntoClasses(t1);
7     cls2 = onto.getOntoClasses(t2);
8
9     If cls1 or cls2 Is Empty Returns 0.0;
10
11    If intersection(cls1, cls2) Is Not Empty Returns 1.0;
12
13    cls_sup = onto.getCommonParentClasses(cls1, cls2);
14    tot_docs = index.totalDocs;
15    If cls_sup Is Not Empty {
16       prob_t1 = index.getFrequency(cls1) / tot_docs;
17       prob_t2 = index.getFrequency(cls2) / tot_docs;
18       prob_tsup = index.getFrequency(cls_sup) / tot_docs;
19       If log(prob_t1 * prob_t2) != 0 {
20         a_term = 2 * log(prob_tsup) / log(prob_t1*prob_t2);
21       }
22    }
23
24    (tot_props1, rel_props1) = onto.countProps(cls1, cls2);
25    (tot_props2, rel_props2) = onto.countProps(cls2, cls1);
26
27    If tot_props1 + tot_props2 > 0 {
28       b_term = (rel_props1 + rel_props2) /
29                         (tot_props1 + tot_props2);
30    }
31
32    Returns (Beta * a_term) + (1-Beta * b_term);
33
34 }
35
```

Figure 6: Algorithm for computing Terms Similarity.

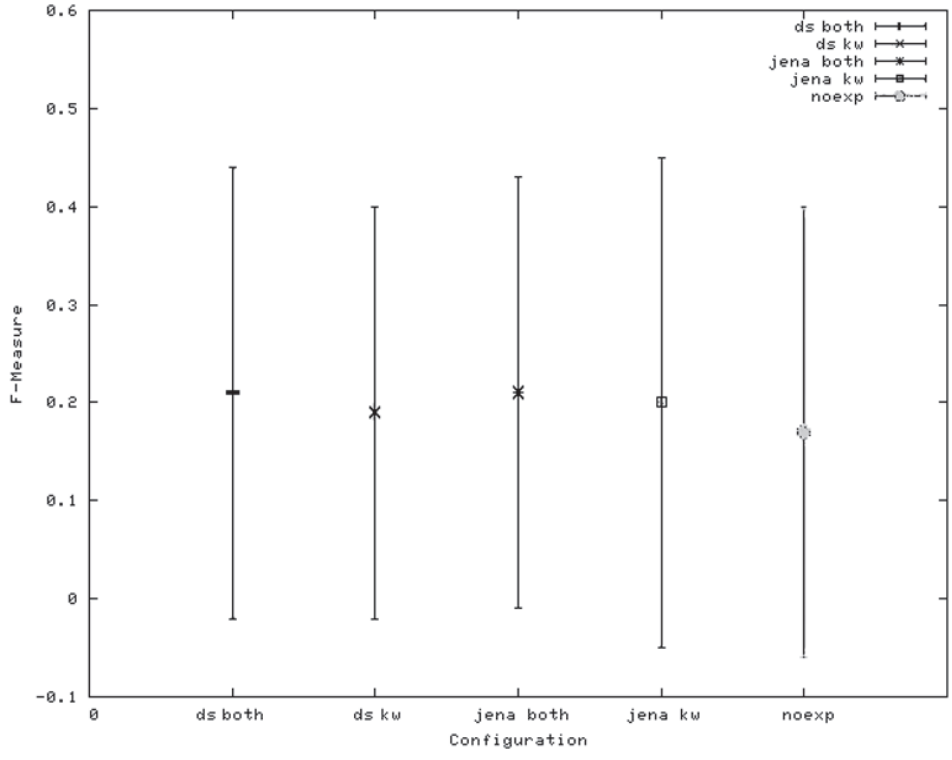

Figure 7: F-Measure for the global tests.

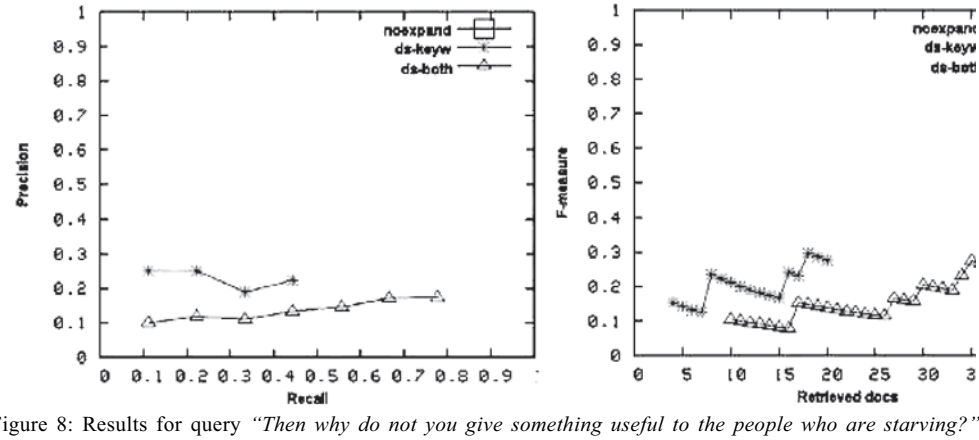

Figure 8: Results for query *"Then why do not you give something useful to the people who are starving?"*. (a) *n* point average precision. (b) Retrieved documents vs. F-measure.

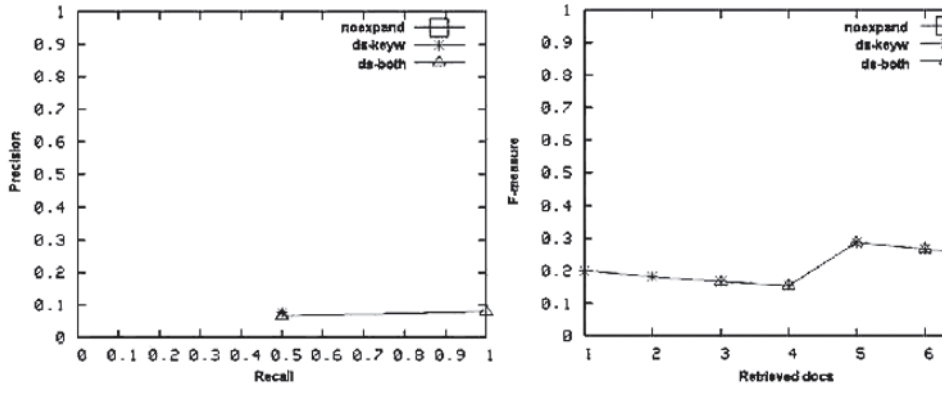

Figure 9: Results for query *"Why do you not give something to people who needs?"*.
(a) *n* point average precision. (b) Retrieved documents vs. F-measure.

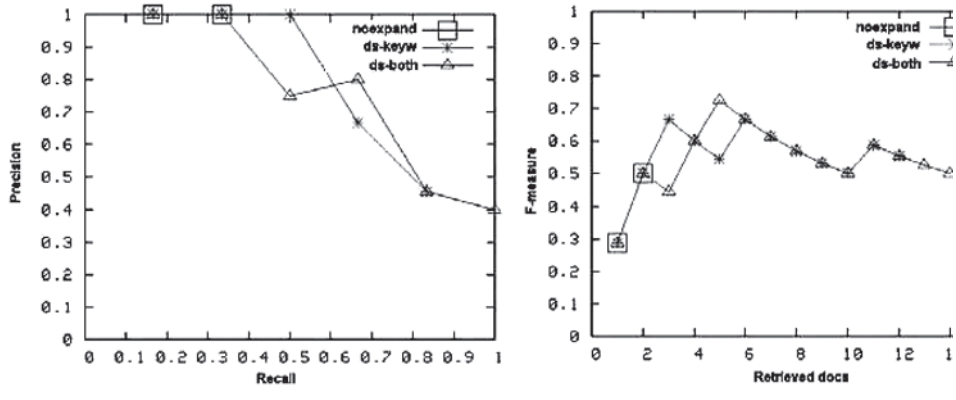

Figure 10: Results for query *"How do you sell your work?"*.
(a) *n* point average precision. (b) Retrieved documents vs. F-measure.

6. Related Works

As mentioned in Section 3, ontologies have been recently used in Information Retrieval (IR). OntoSeek [9] proposed the use of ontologies to increase both recall and precision in narrow domains, such as product catalogs. OntoSeek used a limited language to represent concepts and a large ontology, WordNet [19], for concept matching. Khan proposed, in [13], the ideas of conceptual distance were applied to retrieve audio data using a query expansion mechanism that deals with natural language user queries in the domain of sports. In terms of techniques used for retrieval this is the work most related to ours.

Some work related to automated content-based video retrieval was developed at Cambridge University, and published in [5], which presents a statistical approach for browsing multimedia documents, specifically broadcast news video. Other works use MPEG-7[9] to provide more detailed information about the content of audio or video data, and exploit it to make the retrieval[3].

An use of ontologies for video retrieval was proposed by the ISIS research group from the University of Amsterdam [27], offering an interactive method for video retrieval by guiding the user's interaction with domain information extracted from an ontology.

Another work related to ours is about natural language Question Answering (QA) systems [10]. These systems return specific answers, instead of documents, to questions (usually in natural language) posted by the users. QA is related to IR in the sense that they both extract information from a collection and need to have indexing and retrieval stages. But, QA systems usually concentrate efforts on natural language processing in both stages. We did not use the QA approach because we wanted to retrieve additionally to the response the context in which it is, and we could have more than one answer to the user question. Anyway we think that OnAIR performance can be improved using some QA techniques.

An analysis of the question can be used to restrict the possible answers, as in the Pergunte! system [22].

This system is a Brazilian Portuguese QA system for the web. It uses a pattern-based classifier to associate a category to the user question, and also uses an analysis of morphological category of the terms in the question to match the documents and the candidate fragments to extract the expected answers. The FaqFinder system [6], is a QA system used to query specific information in FAQ (Frequently Asked Questions) databases. In addition to question classification, FaqFinder is concerned with terms disambiguation using Wordnet [19] to disambiguate words in a lexical level.

In January, 2001, The Krannert Art Museum in Champaign, IL, USA, featured a large exhibition on the work of sculptor Jacques Lipchitz (1891-1973). Besides sculptures, drawings, and paintings the show presented a software developed by Bruce Bassett and Histor Systems entitled "Conversartion with Jacques Lipchitz:

A Breakthrough in Interactivity". This system started to be developed in the 1970s and, in its first version, worked with VHS tapes that were manually selected and played according to the asked question.

The version presented at the Krannert Art Museum in 2001 upgraded this pioneering work by using digitized video fragments. However, since it was part of a commercial enterprise we have little information about its implementation. It might have used ontologies. In any case, Bassett's work with the cubist sculptor, which summed up to 300 hours of taped interviews, is certainly a precursor to what we developed. It nicely broke with traditional exhibition practices and provided a very attractive interaction between the artwork and the spectators, bringing back the thoughts on art of the deceased sculptor[14, 4].

7. Conclusions and Future Work

In this paper, we show that the use of an ontology based query expansion mechanism in OnAIR, within the domain of contemporary art using natural language queries, improves in average the system performance in terms of precision and recall.

We implemented OnAIR in a domain-independent manner, so we can use it with other clip collections by changing the underlying ontology and associating keywords (or the speech transcription) for each clip.

This is important because we avoid (or at least minimize) implementation effort to use the video retrieval in other collections.

One promising application of OnAIR is in the educational field, using recorded seminars or courses. This applicability is because usually this kind of material has long duration and is specific to a certain domain.

We are currently working with a course of Design Patterns and Object-Oriented Development lectured by Joseph Yoder at University of São Paulo. We started transcripting the clips, and our next steps are: (1) to identify keywords for each clip and, (2) to develop an ontology that covers all the concepts in the course.

We expect to make the course available through

[9]MPEG-7 is a standard for description and search of audio and visual content.

On-AIR queries, to allow students querying about specific topics and to get this information needing not to browse the entire course[21]. A functional prototype of this course was presented during the Fifth Latin American Conference on Pattern Languages of Programming[10].

An interesting contribution of this work was the use of several tools in a complex application. We used RSLP for stemming, Jazzy API for misspelling correction and Jena for ontology manipulation. Finally we used JMF for video manipulation inside the Java application.

We are planning future work based on some ideas of natural language processing present in question answering systems. In addition, we want to model the dialogue between the user and the system. In the current state of the system, each query is treated independently.

Viewing the sequence of queries as a dialogue may help the system to better understand what kind of information the user is looking for, helping the user to improve his queries and helping the system to understand its information needs. At present the queries are not deeply analyzed and their logical structure is not taken into account. Future work would include natural language processing and detecting the presence of negations, disjunctions or conjunctions in the query.

Acknowledgements

OnAIR development was supported by CAPES. Paula P. Braga is supported by a grant from CNPq, Process # 140923/03-9.

Renata Wassermann is partially supported by the CNPq grant PQ 304486/2004-3. This work has been supported by CNPq project 55.0222/2003-0.

We would like to thank Fabio Kon for his technical review and ideas given to this project and the anonymous referees for the comments.

References

[1] T. Andreasen, J. Nilsson, and H. Thomsen. Ontology-based querying. In *Proceedings of the Fourth International Conference on Flexible Query-Answering Systems*, pages 15–26, Warsaw, Poland, Agosto 2000.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley Longman, 1999.

[3] W. Bailer, H. Mayer, H. Neuschmied, W. Haas, M. Lux, and W. Klieber. Content-based video retrieval and summarization using MPEG-7. In *Proceedings of the Internet Imaging V*, pages 1– 12, San Jose, CA, USA, Janeiro 2004.

[4] B. Bassett and Histor Systems. Conversation with Jacques Lipchitz: A breakthrough in interactivity, 2001. http://www.conversationwithjacqueslipchitz.org/.

[5] M. G. Brown, J. T. Foote, Gareth J. F. Jones, K. Sparck-Jones, and S. J. Young. Automatic content-based retrieval of broadcast news. In *Proceedings of the 3rd ACM Multimedia Conference*, pages 35–43, San Francisco, USA, Novembro 1995.

[6] R. Burke, K. Hammond, V. Kulyukin, S. Lytinen, N. Tomuro, and S. Schoenberg. Natural language processing in the faq finder system: Results and prospects. Technical report, AAAI Spring Symposium, 2002.

[7] World Wide Web Consortium. RDF Resource De.nition Framework, 2004. http://www.w3. org/RDF/.

[8] J. Gennari, M. Musen, R. Fergerson, W. Grosso, M. Crubézy, H. Eriksson, N. Noy, and S.Tu. The evolution of Protégé–2000: An environment for knowledge–based systems development. *International Journal of Human-Computer Studies*, 58(1):89–123, 2003.

[9] N. Guarino, C. Masolo, and G. Vetere. Ontoseek: Content-based access to the web. *IEEE Intelligent Systems*, 14(3):70–80, Maio 1999.

[10] L. Hirschman and R. Gaizauskas. Natural language question answering: the view from here. *Natural Language Engineering*, 7(4):275–300, 2001.

[11] E. Hyvönen, A. Styrman, and S. Saarela. Ontology-based image retrieval. In *Towards the semantic web and web services, Proceedings of XML Finland 2002 Conference*, pages 15–27, Finland, 2002.

[12] P. Jackson and I. Moulinier. *Natural Language Processing for Online Applications: Text Retrieval, Extraction, and Categorization*. John Benjamins Publishing Co, 2002.

[13] L. Khan. *Ontology-based Information Selection*. PhD thesis, Department of Computer Science, University of Southern California, 2000.

[14] G. Kline. High-tech sculptor has the answers. *The News-Gazette Online*, October 2001. Published in WWW in October 2001: http://www.news-gazette.com/story.cfm?Number=10249.

[15] H. Knublauch, M. Musen, and A. Rector. Editing description logics ontologies with the Protégé OWL plugin. In *International Workshop on Description Logics*, Whistler, BC, Canada, 2004.

[16] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, pages 296–304, San Francisco, USA, 1998. Morgan Kaufmann Publishers Inc., 1998.

[17] M. Mauldin. *Conceptual Information Retrieval: A case study in adaptive partial parsing*. Kluwer Academic Publishers, 1991.

[10] http://sugarloafplop2005.icmc.usp.br/

[18] B. McBride. Jena: Implementing the rdf model and syntax specification. In *Proceedings of the Second International Workshop on the Semantic Web*, Hong Kong, China, May 2001.

[19] G. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.

[20] V. Orengo and C. Huyck. A stemming algorithm for the Portuguese language. In *Proceedings of the 8th International Symposium on String Processing and Information Retrieval(SPIRE) 2001*, pages 186–193, 2001. An implementation of the algorithm in C is available at: http://www.cs.mdx.ac.uk/research/PhDArea/rslp/RSLP.htm.

[21] C. Paz-Trillo, R. Wassermann, and F. Kon. A pattern-based tool for learning design patterns. Technical Report RT-MAC-2005-04, Instituto de Matemática e Estatística, Universidade de São Paulo, 2005. Available in: http://www.ime.usp.br/~cpaz/rt-mac-2005-04.pdf.

[22] J. Rabelo. Pergunte! uma interface em português para pergunta-resposta na web. Master's thesis, Informatics Center, Federal University of Pernambuco, Brazil, 2004.

[23] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975.

[24] M. Smith, C. Welthy, and D. McGuiness. OWL Web Ontology Language Guide. Technical report, World Wide Web Consortium, 2004. http://www.w3.org/TR/2004/REC-owl-guide-20040210/.

[25] R. Ueda. Ispell Dictionary for Brazilian Portuguese: br.ispell, 2002. Available at http://www.ime.usp.br/~ueda/br.ispell/.

[26] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, 2nd edition, 1979.

[27] M. Worring, A. Bagdanov, J. v. Gemert, J-M. Geusebroek, M. Hoang, A.Th. Schreiber, C.G.M. Snoek, J. Vendrig, J. Wielemaker, and A.W.M. Smeulders. Interactive indexing and retrieval of multimedia content. In *Proceedings of the 29th Conference on Current Trends in Theory and Practice of Informatics*, volume 2540 of *Lecture Notes in Computer Science*, pages 135–148, Milovy, Czech Republic, 2002. Springer-Verlag,2002.