Supporting multimedia recommender systems with peer-level annotations

Marcelo G. Manzato, Rudinei Goularte
Mathematics and Computing Institute
University of Sao Paulo
Av. Trabalhador Sancarlense, 400
PO Box 668 – 13560-970
Sao Carlos, SP – Brazil
{mmanzato, rudinei}@icmc.usp.br

ABSTRACT

Peer-level annotation stands for the enrichment of content by any user, who acts as author, being able to make annotations, using, for instance, handwriting or speech recognition capabilities. This type of annotation makes users comfortable when taking digital notes, as they do in every day life. This is an advantage over hierarchical authoring, which is a time-consuming task usually employed by content providers. This paper proposes a content-based recommender architecture which explores information that is available at the time users enhance content. This feature enables our architecture to reach a certain level of semantic information from the content and from user's preferences, which is essential for recommender systems applications.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.5 [**Information Interfaces and Presentation**]: Multimedia Information Systems—*Video (e.g., tape, disk, DVI)*

General Terms

Design, Algorithms

Keywords

Recommendation, semantic information, interaction, peer-level annotation, enrichment.

1. INTRODUCTION

One of the greatest advantages that digital television has brought to users is the capability to interact with the content. This affirmative can be supported if we look at the huge advances of the web, whose main characteristic is the interactive scenario where users are able to choose alternative navigation paths, explore different pieces of information and click on related links they wonder to check. The interaction with multimedia content, in special, is obtaining extra development efforts, because of the recent availability of web-based authoring tools, such as YouTube, Facebook, etc. Those tools have changed the user-consumer role into an user-producer role, enabling any user to create multimedia content and make it available on the web.

Usually, the authoring process can be done following two different approaches: hierarchical and peer-level annotation [10]. The hierarchical approach provides information about specific media items with the objective to be searched or analyzed. One example is metadata referring a movie, such as title, producer, list of actors, etc. Cesar et al. [13] argue that this type of information is essential, in interactive television, for content personalization. In addition, they mention that it can be done manually or automatically, and that content providers are responsible to create those metadata.

Different from the hierarchical paradigm, in the peer-level approach the annotation activity is done by any person. One example is the highlight of an actor's text name, or the gesture of cycling an object using electronic ink. One of the characteristics of this level of annotation is that it does not follow a restrictive vocabulary, but provides freedom of expression to the user [13].

As previously mentioned, the authoring process obtains extra attention from users as it allows them to act as content providers and personalize the data, adding valuable metadata as additional information semantically related. Usually, this additional information is not fully explored, remaining at user's storage, or eventually, shared with friends [14][31]. However, different applications require the use of semantic information about the content in order to provide content-related services to users. Furthermore, the extraction of high-level information is difficult to be accomplished, making those applications to face the semantic gap problem [34]. A number of applications which depend on semantic information can be mentioned: object extraction and recognition [35][23], content selection [38], adaptation and personalization [15], recommender systems [4], etc.

Recommender systems, in special, constitute a vast research area as it provides ways to help users to deal with information overload. It also makes available personalized content and services, according to users' interests [4]. YouTube, for instance, provides to the user a list of videos whose metadata matches the ones attached to the content that was being watched. However, two limitations can be mentioned

WebMedia'09, October 5–7, 2009, Fortaleza, CE, Brazil.

in most systems like YouTube: i) sometimes the metadata of a video is not sufficient to describe the whole content, mainly because the author didn't provide a complete description of the content; and ii) the recommender will always select pieces of content without checking whether the user has liked or not the whole or part of the current video. Consequently, it can be noticed that the semantic gap faced by recommender systems cannot be restricted to the content itself, but it needs to be extended to the user's feelings about specific scenes or the whole video content.

The need of semantic metadata to create recommender systems certainly can be supported by hierarchical content authoring, because content providers have the objective in mind to create useful and meaningful information related to the multimedia presentation. However, the literature [1][28] reports that the job of annotating is time-consuming, and requires huge efforts from producers to accomplish metadata creation for lots of multimedia data that is available nowadays. Furthermore, the authoring process of semantic information may explore different aspects of the content. Usually, producers will annotate certain characteristics that he/she subjectively thinks it is important for future applications. One may think that the place where a car is parked is more important than the car's model and brand, which, on the other hand, may be seen as the most interesting piece of information for another.

Consequently, as an attempt to minimize the limitations related to hierarchical content authoring, the exploration of peer-level annotation may bring interesting results when applied to the problems mentioned above. Firstly, the time-consuming question can be partially solved if multiple users, including those which are not professionals from providers centers, dedicate collaborative effort to the same content. As previously exposed, ordinary users like to act as content producers, and this fact may be explored for semantic information extraction activities. Secondly, the subjective choice of different aspects of the content to be annotated has a slighter drawback if we consider that the information extracted by the user interaction will be used for personalization services, which, in turn, will benefit the same user or a set of buddies that usually shares the same preferences.

However, if peer-level annotation can contribute to minimize the problems that are inherent to hierarchical content authoring, on the other hand it brings challenges that need further research. One of them is the fact that interaction between user and content can be done in different ways. Thus, a number of techniques must be available in order to analyze the interactive scenario and find valuable information that can be considered metadata. Another challenge is the peer-level characteristic of not following a restrictive vocabulary; consequently, algorithms to convert interaction-based extracted data into a representative format must be developed.

In a previous work [24], we have started a research to support metadata extraction by exploring peer-level annotations. The proposed technique generates an user's profile based on his/her interests about specific subjects of pieces of news. A representation model was also proposed in order to minimize the unrestrictive annotation vocabulary mentioned by Cesar et al. [13]. However, a number of topics were not explored by the technique: i) the use of peer-level annotation exploration tools in different domains, such as movies; ii) the combination of different interaction paradigms, such as user's speech, annotation, regions of interests, captured frames and handwriting; and iii) the provision of an application which explores the generated user's profile.

Considering all issues we have just depicted, this paper proposes an enchanced architecture to support the topics mentioned above which were not explored previously. Specially, we propose a recommender system technique based on peer-level annotation which is able to provide to the user related content according to the video being watched, and also, according to the user's feelings about specific scenes. To accomplish this, we use a set of techniques to extract metadata from the content and also from user interaction, such as speech and handwriting recognitions, captured frames, closed-caption and face detection and recognition.

This paper is organized as following. Section 2 presents the related work of this paper, which is divided into three subareas: content enrichment, metadata extraction and recommender systems. Section 3 depicts the video description pre-processing step and a set of peer-level-based metadata extraction techniques. Section 4 proposes the recommender architecture, which is based on user's enrichment activity. Section 5 describes the experimental results by presenting two use scenarios. Section 6 depicts the final remarks, including the future work; and finally section 7 presents the acknowledgments of this paper.

2. RELATED WORK

This section depicts the related work, which is divided in this paper into three main subareas: content enrichment, metadata extraction and recommender systems.

Content enrichment can be accomplished using different interaction paradigms, and its purpose may vary according to user's intentions. Usually, to provide a pleasant enrichment tool to the user, the interaction follows the natural interfaces concept, previously defined by Abowd & Mynatt [2] as a subarea of ubiquitous computing [37]. Thus, there is a variety of work [31][20] which simulates the action of handwriting into a piece of content, using, for instance, a pen-based device like a tablet augmenter or a touchscreen mobile phone. Regularly, strokes produced by the user are stored locally or shared with friends [14], and later, they can be used to create personalized multimedia presentations [9].

In addition to the use of handwriting interaction paradigm, some authors explore other ways of enrichment. Speech, for instance, can be used to infer certain information of user's feeling, as described in [39], where the content is classified into a set of categories, such as neutral, anger, hapiness and sadness. The use of images, in turn, is explored by Boll et al. [7] to support the creation of photos albums using content-based and contextual metadata. This metadata is applied into a set of pre-defined rules in order to collect and share related documents and/or photos from the Web to enrich personal albums.

When dealing with metadata extraction, hierarchical authoring is a time-consuming task, and hence, should not be delegated only to content providers, as more content will be available each day [36][11]. Consequently, the insertion of consumers into the metadata authoring process is a interesting design plan, being mentioned by many work available on the literature.

Some of those work try to model the feelings which are supposed to be present on the user when watching specific scenes [22][18]. If, on the one hand, it is possible to extract certain metadata, on the other hand, it has some limitations, because it is not possible to generalize expected user emotions for specific multimedia content. Other research deals with Integrated Media Capture Environment (IMCE) [3] and personal traits [5]; however, there are some drawbacks in these approaches, such as the hard procedure to extract metadata from third-party multimedia content, and the not trivial task of mapping between user's preferences and audiovisual content.

The third subarea depicted in this section is about recommender systems. Usually, techniques adopt one of the following approaches [6]: i) content-based recommendations [6][30][32], where the user will be recommended items similar to the ones the user preferred in the past; ii) collaborative recommendations [8][16], where content selection is based on similar tastes from other people; and iii) hybrid approaches [6][29], which combine collaborative and content-based recommendations.

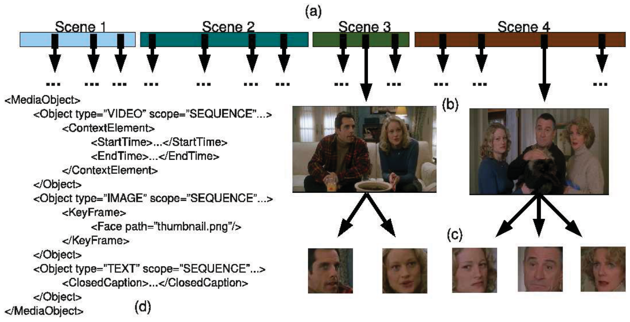
Recommendations are based on a rating of pieces of content, which is defined by the users on a subset of videos that they have already seen. After this initial input that gives information about user's interests, the system searches related content according to different approaches – most of them exploring textual information: term frequency measure [32], Bayesian classifiers [27], clustering, decision trees and artifical neural networks [30], etc. The main limitations of the state-of-art techniques are: i) the lack of efficient content analysis algorithms, which makes difficult the extraction of semantic information by analyzing audiovisual information; ii) overspecialization, which makes the recommender to select pieces of content whose subject has already seen by the user; and iii) the new user problem, which requires from new users at least some ratings before recommending additional content [34].

At this point, one can note that the three subareas depicted in this section have the user as the most important element of the whole system. Thus, it is worth to consider joining them as an attempt to solve the challenges inherent to each subarea, as described earlier in this section. This paper proposes a recommender system which explores metadata extracted from user enhancement activity. The main advantages of this approach are: i) the exploration of users' additional data, not requiring from them boring and time-consuming efforts to create metadata, once they will be annotating personal information without worrying about the true application's intention behind the interface; and ii) the provision of recommendations based on content semantics and users' feelings about specific scenes of generic domain video streams.

3. ANNOTATION AND METADATA EXTRACTION

The recommender system proposed in this paper requires video strems to be prepared before using them as multimedia database. Furthermore, a set of techniques must be available in order to extract metadata at the time when users will be augmenting the content. This section describes the pre-processing steps which will give subsidies to recommend related audiovisual data.

3.1 Video Description



```
<MediaObject>
    <Object type="VIDEO" scope="SEQUENCE"...>
        <ContextElement>
            <StartTime>...</StartTime>
            <EndTime>...</EndTime>
        </ContextElement>
    </Object>
    <Object type="IMAGE" scope="SEQUENCE"...>
        <KeyFrame>
            <Face path="thumbnail.png"/>
        </KeyFrame>
    </Object>
    <Object type="TEXT" scope="SEQUENCE"...>
        <ClosedCaption>...</ClosedCaption>
    </Object>
</MediaObject>
```

Figure 1: Semi-automatic video description procedure.

Figure 1 illustrates the video description procedure, which is, in this paper, a semi-automatic task. It starts at shot boundary detection or temporal segmentation. Some authors suggest a video stream to be divided into "scenes" (Figure 1 (a)), and a scene, composed by one or more camera "shots" [33]. Thus, a shot is the separation of basic video units representing continuous action in both time and space into a scene. A number of techniques is available to detect shot boundaries [25][12]; in this paper, we use a previous proposed technique [25] which is based on intelligent systems.

After the shot boundary detection, the next step is responsible for clustering adjacent shots with the same meaning, in order to obtain a scene-structured video stream. This procedure is done manually, but, as soon as we have the beginning and ending timestamps of each scene, we use an implemented library to extract the corresponding subtitle from SubRip (SRT) files.

In addition to the extraction of subtitle, we select various keyframes from each scene (Figure 1 (b)). This process is also done manually, and its heuristic is to choose frames that visibly show the face of any person.

The last step of video description is responsible to apply a face detection algorithm into each selected keyframe in order to check if there is a person into the scene. We use the Face Annotation Interface Java API (faint)[1], that extracts faces and stores them locally as thumbnails, together with the person's face name, which is obtained from user interaction.

It is important to note that all metadata extracted from this video analysis is represented in XML documents, which follow the MediaObject representation model [21]. The MediaObject provides mechanisms to describe multimedia content in an organized way, separating each type of media descriptions (structural, compositional, context and linking) in a set of MPEG-4 objects with related MPEG-7 descriptions. Figure 1 (d) illustrates some code to represent scene boundaries, detected faces, keyframes and related subtitle.

3.2 Peer-level Enrichment

The previous subsection described the video description procedure, which is executed with all video streams stored into the multimedia database; it makes available the search of scenes according to their content. This subsection, in turn, depicts how metadata is extracted from user enrichment in order to be used as user's preferences information to retrieve related pieces of content.

[1] http://faint.sourceforge.net/

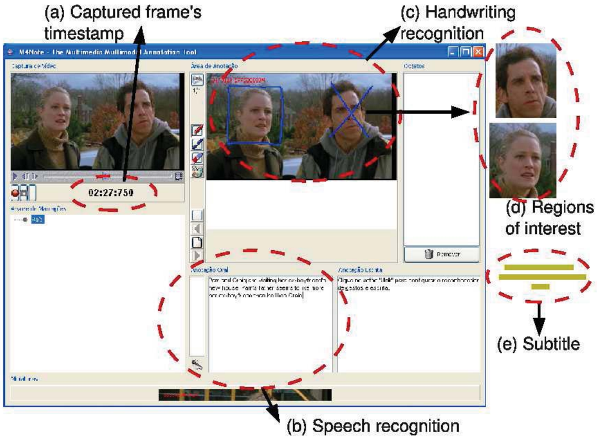


Figure 2: Peer-level enrichment environment using the M4Note application: at the upper left-hand corner is the video visualization panel; and at the upper right-hand corner is the electronic ink edition area the selected frames. Dashed ellipses represent extracted metadata that describes user's preferences.

Figure 2 illustrates the M4Note tool [19], which is based on multimodal interfaces with electronic ink and voice, providing the user with annotation mechanisms over a multimedia object stream. Implemented as a capture and access application [2], the tool allows the annotation task to be carried out during a live experience, using a portable device, like a Tablet PC. In this paper, exclusively, M4Note was adapted in order to receive incoming streams from content providers.

The metadata extraction techniques explore different interaction paradigms; these are represented in Figure 2 with dashed ellipses and corresponding captions. Figure 2 (a) references the captured frames' timestamps, used in our system to delimitate a period of subtitle which will be used (and explained later) as keywords for searching procedures.

Figure 2 (b) references the speech recognition module: users are allowed to comment about captured frames, and consequently, all comments are transformed to text, and also used later as keywords for searching procedures. In this case, the keywords represent user's preferences and feelings about what he/she is annotating.

Figure 2 (c) references the handwriting recognition module. A technique was integrated into the tool, which is able to receive as input a set of points, and output the corresponding text or symbols recognized. Our system uses user-defined symbols in order to infer the user's preferences and feelings about the content. So, when the user draws an 'x', for instance, it may imply that the content is not relevant; and when the user draws a 'square' symbol, it may imply that it is relevant.

The set of symbols recognized by the handwriting module is further used to define regions of interests (ROIs) from the captured frame. Figure 2 (d) represents the ROIs extracted from the symbols annotated into the frame: they are defined by the horizontal and vertical maximum and minimum coordinates of each stroke made by the user.

Finally, Figure 2 (e) represents the keywords defined from the subtitle, which was delimitated by the captured frames' timestamps. The next section describes how this set of words is used to make available recommendations based on content semantics.

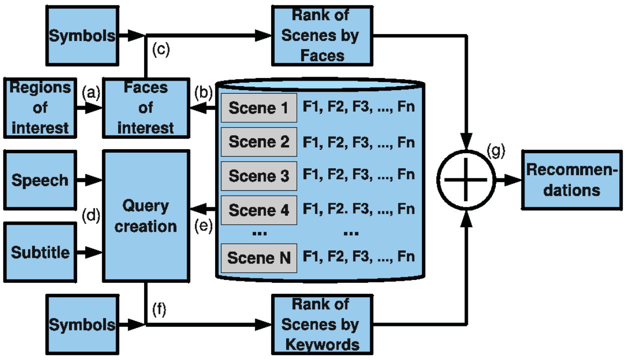

Symbols
(c)
Rank of Scenes by Faces
Regions of interest
(a)
Faces of interest
(b)
Scene 1 F1, F2, F3, ..., Fn
Scene 2 F1, F2, F3, ..., Fn
Scene 3 F1, F2, F3, ..., Fn
Scene 4 F1, F2. F3, ..., Fn
...
...
Scene N F1, F2, F3, ..., Fn
(g)
Recommendations
Speech
(d)
Query creation
(e)
Subtitle
Symbols
(f)
Rank of Scenes by Keywords

Figure 3: Proposed architecture for peer-level-based recommender system.

All information generated by the M4Note tool, such as annotation's points, recognized symbols, ROIs, captured frames and their timestamps are also described and stored according to the MediaObject model [21]. For more details about the M4Note tool, readers should refer to [19].

This subsection presented a set of metadata extraction techniques based on peer-level annotation. These techniques are used in this paper to provide recommendations according to content semantics, user's preferences and feelings. Additional metadata extraction techniques descriptions can be found in [24].

4. RECOMMENDER SYSTEM

Based on the peer-level annotation and metadata extraction techniques that were depicted in the previous section, we propose in this section an architecture that gives support to the exploration of interaction capabilities among user and content. Figure 3 presents the overall schema. The system receives as input data the regions of interests, speech recognized text and a period of subtitle from the content that has been watched; all of them extracted from user annotations, whose methods were described in subsection 3.2. The recommender architecture is composed of two main modules that create intermediary ranks of scenes according to specific types of information.

The first module is responsible to use the defined regions of interest to detect possible faces of interest (Figure 3 (a)). Detected faces are then recognized (Figure 3 (b)), using a database of pre-recognized faces, which was previously created at video description time, as described in subsection 3.1. Persons' faces are then automatically classified into interesting or not, using the annotated symbols (Figure 3 (c)), whose meaning the user should be aware of. Then, all this information is used to create a rank of scenes by faces (simply rank of faces), which is further explored by the architecture.

The second module is responsible to use recognized speech annotations, as well as a period of subtitle defined by the captured frame's timestamp (Figure 3 (d)). This set of words is used to query scenes (Figure 3 (e)), which are later also classified according to the symbols made into the corresponding frame (Figure 3 (f)). Finally, a rank of scenes by keywords (simply rank of keywords) is created, which is combined with the rank of faces (Figure 3 (g)), generating a set of recommendations based on user's interests.

In the following subsections, we depict in more details the creation of both intermediary ranks; the combination of them to generate the recommendations is trivial, as we just calculate the mean between the ranks of each scene. Additional combination mechanisms will be explored in future work.

4.1 Rank of Faces

```
Input: Set of scenes from database
Input: Set of ROIs from annotation
Output: Rank of faces
foreach scene s from database do
    $rank_s^f = 0.1$;
end
foreach ROI r extracted from annotation do
    $F_r = detectFace(r)$;
    foreach detected face f from $F_r$ do
        $p = recognizeFace(f)$;
        foreach scene s from database do
            if $r.symbol == $ 'square' then
                $rank_s^f \mathrel{+}=$
                $p.weight * \frac{numberFaces(s,p.name)}{numberFaces(s,null)}$;
            end
            else
                if $r.symbol == $ 'x' then
                    $rank_s^f \mathrel{/}= 2$;
                end
            end
        end
    end
end
return $rank^f$;
```

Algorithm 1: Algorithm for constructing rank of faces

The rank of faces is created as illustrated in Algorithm 1. It receives as input the set of all described scenes stored in the database, and also, the set of ROIs which was generated from the coordinates of the symbols made by the user.

For each region of interest r extracted from annotation (as illustrated in Figure 2 (d)), we use the faint API to detect possible faces. As the user is able to annotate into more than one face at time, the algorithm has an inner loop, which goes through all faces from the same ROI. For each face f detected, the system tries to recognize it against a set of pre-recognized faces stored into the database. This step is represented by the function $recognizeFace(f)$ in Algorithm 1, which returns to the variable p a data structure composed of the person's face name, and a weight of how close the recognized face is when compared to the samples stored in the database.

The following procedure, as illustrated in Algorithm 1, is responsible to update the rank of faces according to the recognized person. It depends on the symbols made by the user, so that it is possible to infer that the current face is relevant or not. Although we can use different symbols, in this paper we represent both situations using only two, respectively: 'square' and 'x'. In the first situation, the rank of scene s is updated as follows:

$$rank_s^f + = p.weight * \frac{numberFaces(s, p.name)}{numberFaces(s, null)} \quad (1)$$

where $p.weight$ is the weight of how close the face was recognized; $numberFaces(s, p.name)$ returns the number of times person $p.name$ appears in all faces detected in scene s; and $numberFaces(s, null)$ returns the number of faces detected in scene s.

In the second situation, as the user explicitly shows his/her lack of interest on the region, we simply penalize the rank of scene s when dividing it by two.

4.2 Rank of Keywords

```
Input: Set of scenes from database
Input: Text $a$ from speech
Input: Text $b$ from subtitle
Output: Rank of keywords
$speechStr =$ "";
$subtitleStr =$ "";
foreach captured frame c from annotation do
    $hasSquare = false$;
    $hasX = false$;
    foreach ROI r extracted from annotation do
        if $r.symbol == $ 'square' then
            $hasSquare = true$;
        end
        else
            if $r.symbol == $ 'x' then
                $hasX = true$;
            end
        end
    end
    if ((!hasSquare and !hasX)or(hasSquare and
    !hasX)or(hasSquare and hasX)) then
        $speechStr \mathrel{+}= truncate(a_c)$;
        $subtitleStr \mathrel{+}= truncate(b_c)$;
    end
end
$query = createQuery(speechStr, subtitleStr)$;
$rank^k = retrieveLSI(query)$;
return $rank^k$;
```

Algorithm 2: Algorithm for constructing rank of keywords

The rank of keywords is created as illustrated in Algorithm 2. It receives as input the set of all described scenes stored in the database, and also, the sets of speech and subtitle which are related to each captured frame. We have used a period of P seconds before and after the captured frame's timestamp in order to delimitate the used text from subtitle.

The first procedure, as show in Algorithm 2, is to prepare the texts by filtering each statement using the symbols made by the user into the corresponding captured frame. In addition, the function $truncate(.)$ removes eventual stopwords and stems the remaining terms[2]. The former has the objective of removing words that cannot carry significant information, such as adverbs, articles, and linking words. The latter, on the other hand, has the objective of creating unique words that have the same radical. This pre-processing step is necessary to minimize noise that may negatively influence the results.

Following the algorithm, the next procedure is to create the query statement, represented in Algorithm 2 by the

[2] In this paper, we use the Portuguese Porter Stemmer, available at: http://snowball.tartarus.org

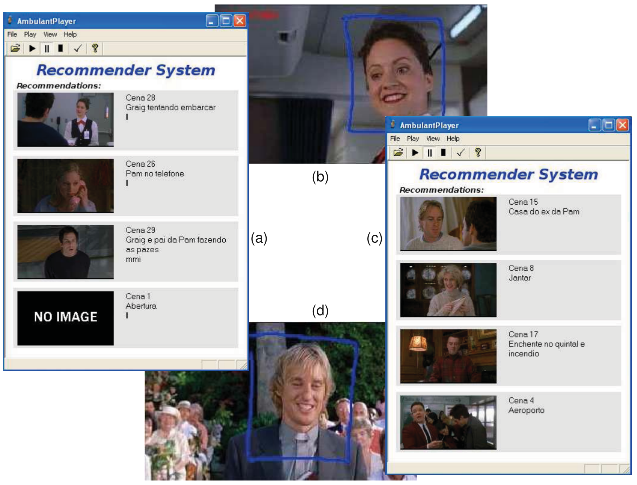


Figure 4: Experimental results for two use scenarios. The SMIL presentations (a) and (c) were generated based on user annotations (b) and (d), respectively.

function $createQuery(speechStr, subtitleStr)$. This function consists of filling a query buffer of size Q, selecting terms from filtered speech and subtitle, according to the three-rules heuristic:

1. **Priority 1:** Select terms from speech which have an equivalent in subtitle;
2. **Priority 2:** Select remaining terms from speech;
3. **Priority 3:** Select terms from subtitle.

After the creation of the query string, it is submitted to the Latent Semantic Indexing (LSI) [17] module, as represented in Algorithm 2 by the function $retrieveLSI(query)$, which returns a rank of all stored scenes according to the defined keywords.

The LSI technique starts with the load of all scenes' subtitle (referred as documents) stored into the database, following the stopwords removal and stemming procedures. Then, the term-document matrix (TDM) is generated, which is then decomposed into three new matrices by using the Singular Value Decomposition method. After this mathematical process, the matrices' dimensionality is reduced (according to the parameter named k), in order to eliminate not important terms and noise. Finally, the query statement may be used to rank the documents (scenes) according to terms similarities. The detailed descriptions of these steps inherent to LSI can be found in previous work [26].

5. EXPERIMENTAL RESULTS

This section provides the results of the architecture proposed in this paper. It consists of simulating two user annotations into a movie, and then, presenting to the user a set of recommendations which was generated based on the user's interests.

The user annotations follow two scenarios. The first one supposes the user is watching the movie "Meet the Fockers" (2004), which is a sequel to "Meet the Parents" (2000). At the time when Greg and Pam are inside the airplane, and the flight attendant asks Greg to keep his hand luggage, the user stops the movie, captures a frame showing the flight attendant, and makes a square using electronic pen around her head (Figure 4 (b)). At the same time, the user makes a speech annotation, saying: *"Greg now is afraid to give her his hand luggage because last time it didn't passed through the X-Ray, and the airport company lost it during the flight".*

The second scenario supposes the user is watching the same movie, but he/she stops the movie when it shows Pam's ex-boyfriend at the Greg and Pam's wedding scene. A square annotation is also made around his head (Figure 4 (d)), and the user says: *"Kevin will pray in the wedding because he became a rabbi after having known Greg, who is jew".*

The two scenarios consist of recommending scenes from the first movie which are semantically related to the annotations made on the sequel. Thus, for the first case, the system should retrieve scenes related to airport in general, and also, scenes showing the same flight attendant. In the second scenario, in turn, it should retrieve scenes related to Pam's ex-boyfriend, wedding, and also, scenes presenting the fact that Greg is jew.

The parameters used in this evaluation were subjectively defined based on experiments. We used a period $P = 5$ to delimitate the text from subtitle; a buffer $Q = 10$ to set the number of terms in the query string; and $k = 10$ to be used in the dimensionality reduction step by the LSI technique.

Table 1 shows the set of recommendations for both scenarios using: i) only the rank of faces; ii) only the rank of keywords; and iii) both ranks of faces and keywords. The results for scenario 1 show that the recommender system was able to retrieve semantically related scenes according to the information provided by the user. When using only the rank of faces, it returned scene 28 at first place, which is the scene that shows Greg interacting with the flight attendant at departure room and inside the airplane. When using only the rank of keywords, the recommender system retrieved scenes 26 and 29 at first and second places, respectively; the former shows Pam on the phone talking to Greg's answering machine, and mentioning that right now he is flighting; the latter shows Pam's father rescuing Greg after he discussed with the flight attendant about his hand luggage inside the airplane. When using the ranks of faces and keywords, it can be noted that the system merged both results, recommending scenes related to the annotations made by the user. A SMIL presentation was automatically generated from this result, as can be seen in Figure 4 (a).

The results for scenario 2 were also satisfactory. When using only the rank of faces, the system retrieved scenes 15 and 17 at first and second places, respectively. These are the only scenes that show Pam's ex-boyfriend: the former is the visit of Greg and Pam to his new house, and the latter is when Pam's ex-boyfriend brings the altar to her house. When using only the rank of keywords, the system retrieved scene 8 at first place, which is exactly the scene where Pam tells her father that Greg is jew. When using both ranks, the system merged the two previous recommendations, enhancing the retrieved results according to the user's interests. Again, a SMIL presentation was automatically generated from this result, as can be seen in Figure 4 (c).

6. FINAL REMARKS

This paper presented a recommender system architecture based on peer-level annotation which is able to provide to the

Table 1: Recommendations

	Scenario 1	Scenario 2
Faces	28, 29, 3, 26	15, 17, 32, 6
Keywords	26, 29, 1, 3	8, 4, 29, 25
Both	28, 26, 29, 1	15, 8, 17, 4

user related content according to the video being watched, and also, according to the user's preferences about specific scenes. One advantage of the adopted approach is that users can feel comfortable when creating metadata, because their main focus will be on the content enrichment. Another advantage is that the semantic information associated to the annotations provides ways to describe more precisely the user's preferences, considering that the annotations are made by himself.

In this paper, we have also presented the video description procedure, which is an important pre-processing step in order to prepare video streams to be searched by the recommender system. In addition, various techniques related to peer-level annotation were presented, whose objective is to extract metadata from user's enrichment interaction.

Considering the three subareas depicted in the related work of this paper (content enrichment, metadata extraction and recommender systems), we believe that joining them has brought the following contributions:

- Ways to explore additional information that is added by the user at the time he/she is enriching the content;
- The insertion of the user into the metadata creation process, without making this activity a time-consuming and boring task;
- The possibility of using content-based analysis in order to provide recommendations according to content semantics and user's preferences.

As future work, we plan to consider other methods that can be used to explore different interaction aspects, such as tagging systems, sound information (e.g. user's excitement, mood and speech) and assistive segmentation. Also, we plan to compare our recommendation results with other techniques, mainly those which are not based on peer-level annotations. In order to collect substantial evaluation results from our system, we will use real end-users to annotate into the content, and then analyze if the provided recommended content is in accordance with their preferences. In addition, we plan to focus our research on some business model issues, such as: a) who will be benefited by the annotations made by one specific user? b) which content will be recommended to the users? c) what happens if an user does not provide any annotation? and d) should annotations from others be used to improve a specific recommendation? We think that the answers for these questions will contribute to mature the field of recommending content by analyzing user's enrichment information.

7. ACKNOWLEDGMENTS

This work was sponsored by UOL (www.uol.com.br), through its UOL Bolsa Pesquisa program, process number 20090-205103800. The authors would like to thank the financial support from UOL and FAPESP.

8. REFERENCES

[1] G. D. Abowd, M. Gauger, and A. Lachenmann. The Family Video Archive: an annotation and browsing environment for home movies. In *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 1–8, 2003.

[2] G. D. Abowd and E. D. Mynatt. Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction*, 7(1):29–58, 2000.

[3] B. Adams and S. Venkatesh. An embedded suggestive interface for making home videos. In *Proceedings of Int. Conf. Multimedia and Expo*, Toronto, Canada, 2006.

[4] G. Adomavicius and A. Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[5] L. Agnihotri, J. Kender, N. Dimitrova, and J. Zimmerman. User study for generating personalized summary profiles. In *Proceedings of Int. Conf. Multimedia and Expo*, pages 1094–1097, 2005.

[6] M. Balabanovic and Y. Shoham. Fab: Content-Based, Collaborative Recommendation. *Comm. ACM*, 40(3):66–72, 1997.

[7] S. Boll, P. Sandhaus, and U. Westermann. Semantics, Content, and Structure of Many for the Creation of Peronal Photo Albums. In *Proceedings of MM'07*, pages 641–650, 2007.

[8] J. S. Breese, D. Heckerman, and C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proceedings of the 14th. Conf. Uncertainty in Artificial Intelligence*, 1998.

[9] D. C. A. Bulterman. Using SMIL to Encode Interactive, Peer-Level Multimedia Annotations. In *Proceedings of DocEng'03*, pages 32–41, 2003.

[10] D. C. A. Bulterman. Animating Peer-Level Annotations Within Web-Based Multimedia. In *7th Eurographics Workshop on Multimedia*, pages 49–57, 2004.

[11] D. C. A. Bulterman. Is it time for a moratorium on metadata? *IEEE Multimedia*, 11(4):10–17, 2004.

[12] Z. Cernekova, I. Pitas, and C. Nikou. Information theory-based shot cut/fade detection and video summarization. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(1):82–91, January 2006.

[13] P. Cesar, D. C. A. Bulterman, and A. J. Jansen. An Architecture for End-User TV Content Enrichment. *Journal of Virtual Reality and Broadcasting*, 3(9), 2006.

[14] P. Cesar, D. C. A. Bulterman, and A. J. Jansen. Social Sharing of Television Content: An Architecture. In *Ninth IEEE International Symposium on Multimedia*, pages 145–150, 2007.

[15] S. F. Chang and A. Vetro. Video Adaptation: Concepts, Technologies, and Open Issues. *Proceedings of IEEE*, 93(1):148–158, 2005.

[16] W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to Order Things. *J. Artificial Intelligence Research*, 10:243–270, 1999.

[17] S. Deerwester, S. T. Dumais, T. Landauer, G. Furnas, and R. Harshman. Indexing by latent semantic analysis. *Journal of the Society for Inf. Science*, 41(6):391–407, 1990.

[18] N. Eagle. *Machine perception and learning of complex social systems*. PhD thesis, Massachusetts Inst. Technol., 2005.

[19] R. Goularte, J. A. Camacho-Guerrero, V. R. Inacio Jr., R. G. Cattelan, and M. G. C. Pimentel. M4Note: a Multimodal Tool for Multimedia Annotations. In *Proceedings of WebMedia & La-Web 2004 Joint Conference – 10th Brazilian Symposium on Multimedia and the Web & 2nd Latin American Web Congress (La-Webmedia 2004)*, 2004.

[20] R. Goularte, R. G. Cattelan, J. A. Camacho-Guerrero, V. R. Inacio Jr., and M. G. C. Pimentel. Interactive Multimedia Annotations: Enriching and Extending Content. In *Proceedings of DocEng'04*, pages 84–86, 2004.

[21] R. Goularte, M. G. C. Pimentel, and E. S. Moreira. Context-Aware Support in Structured Documents for Interactive-TV. *Multimedia Systems*, 11(4):367–382, 2006.

[22] A. Hanjalic and L. Q. Xu. Affective video content representation and modeling. 7(1):143–154, 2005.

[23] Y. Li, S. Narayanan, and C. C. J. Kuo. Content-Based Movie Analysis and Indexing Based on AudioVisual Cues. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(8):1073–1085, 2004.

[24] M. G. Manzato, D. B. Coimbra, and R. Goularte. Multimedia Personalization Based on Peer-level Annotation. In *Proceedings of the 7th. European Interactive TV Conference (EuroITV'09)*, pages 57–66, Leuven, Belgium, 2009.

[25] M. G. Manzato and R. Goularte. Shot boundary detection based on intelligent systems. In *Proceedings of the XIII Brazilian Symposium on Multimedia and the Web (WebMedia'07)*, pages 190–197, Gramado, RS, 2007.

[26] M. G. Manzato and R. Goularte. Video News Classification for Automatic Content Personalization: A Genetic Algorithm Based Approach. In *Proceedings of the XIV Brazilian Symposium on Multimedia and the Web (WebMedia'08)*, pages 1–8, Vila Velha, ES, 2008.

[27] R. J. Mooney, P. N. Bennett, and L. Roy. Book Recommending Using Text Categorization with Extracted Information. In *Proceedings of Recommender Systems Papers from 1998 Workshop, Technical Report*, 1998.

[28] S. N. Patel and G. D. Abowd. The ContextCam: Automated Point of Capture Video Annotation. 3205:301–318, 2004.

[29] M. Pazzani. A Framework for Collaborative, Content-Based, and Demographic Filtering. *Artificial Intelligence Rev.*, pages 393–408, 1999.

[30] M. Pazzani and D. Billsus. Learning and Revising User Profiles: The Identification of Interesting Web Sites. *Machine Learning*, 27:313–331, 1997.

[31] M. G. C. Pimentel, R. Goularte, R. G. Cattelan, F. S. Santos, and C. Teixeira. Enhancing multimodal annotations with pen-based information. In *Ninth IEEE International Symposium on Multimedia*, pages 207–212, 2007.

[32] G. Salton. *Automatic Text Processing*. Addison-Wesley, 1989.

[33] A. F. Smeaton. Techniques Used and Open Challenges to the Analysis, Indexing and Retrieval of Digital Video. *Information Systems*, 2007.

[34] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[35] X. Song and G. Fan. Joint Key-Frame Extraction and Object Segmentation for Content-Based Video Analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(7):904–914, 2006.

[36] S. Venkatesh, B. Adams, D. Phung, C. Dorai, R. G. Farrell, L. Agnihotri, and N. Dimitrova. "You Tube and I Find" – Personalizing Multimedia Content Access. 96(4):697–711, 2008.

[37] M. Weiser. The Computer of the 21st Century. *Scientific American*, 265(3):94–104, 1991.

[38] Z. Xiong, X. S. Zhou, Q. Tian, Y. Rui, and T. S. Huangm. Semantic retrieval of video - review of research on video retrieval in meetings, movies and broadcast news, and sports. *IEEE Signal Processing Magazine*, 23(2):18–27, 2006.

[39] F. Yu, E. Chang, Y.-Q. Xu, and H.-Y. Shum. Emotion Detection from Speech to Enrich Multimedia Content. *Lecture Notes in Computer Science*, 2195/2001:550–557, 2001.