Discovering similarities for content-based recommendation and browsing in multimedia collections

Taras Lehinevych,[*] Nikolaos Kokkinis-Ntrenis,[†] Giorgos Siantikos,[‡]
A. Seza Doğruöz,[§] Theodoros Giannakopoulos[‡] and Stasinos Konstantopoulos[‡]
[*]*Faculty of Computer Science, National University of "Kyiv-Mohyla Academy", Kyiv, Ukraine*
Email: tleginevych@gmail.com
[†]*Department of Information and Telematics, Harokopio University of Athens, Greece*
Email: nikoskokkini@gmail.com
[‡]*Institute of Informatics and Telecommunications, NCSR "Demokritos", Athens, Greece*
Email: {siantikosg,tyiannak}@gmail.com, konstant@iit.demokritos.gr
[§]*Netherlands Institute for Advanced Studies, Wassenaar, the Netherlands*
Email: a.s.dogruoz@gmail.com

***Abstract*—The purpose of the research described in this paper is to examine the existence of correlation between low level audio, visual and textual features and movie content similarity. In order to focus on a well defined and controlled case, we have built a small dataset of movie scenes from three sequel movies. In addition, manual annotations have led to a ground-truth similarity matrix between the adopted scenes. Then, three similarity matrices (one for each medium) have been computed based on Gaussian Mixture Models (audio and visual) and Latent Semantic Indexing (text). We have evaluated the automatically extracted similarities along with two simple fusion approaches and results indicate that the low-level features can lead to an accurate representation of the movie content. In addition, the fusion approach seems to outperform the individual modalities, which is a strong indication that individual modules lead to diverse similarities (in terms of content). Finally, we have evaluated the extracted similarities for different groups of human annotators, based on what a human interprets as similar and the results show that different groups of people correlate better with different modalities. This last result is very important and can be either used in (a) a personalized content-based retrieval and recommender system and (b) in a "local" weighted fusion approach, in future research.**

***Keywords*-movie recommendation; multimedia signal analysis; audio features; visual features; optical flow; fusion; similarity; recommender systems**

I. Introduction

Recommender systems (RS) aim to provide suggestions regarding information items to users, through predicting user preferences on these items [1], [2]. They can be considered as an application of *pattern analysis* to the task of generating personalized recommendations for several types of information items (e.g. web pages, movies, music, etc.) Depending on the kind of information that forms the basis for these recommendations, we classify recommender systems into:

- *collaborative* (also known as *collaborative filtering*), where recommendations for a specific item and user are estimated from how users with preferences and tastes similar to the specific user rate the specific item. Very roughly speaking, this amounts to recommending an item to a user because other users have rated other items similarly to this user and have rated this item positively; and
- *content-based* [3], where the recommendations are made based on commonalities between the features of similarly rated items. Again very roughly speaking, this amounts to recommending an item to a user because it is similar to other items this users has rated positively.

Besides this basic dichotomy, there are also *hybrid* systems that combine collaborative and content-based recommendation methods [4] and *context-aware* systems that additionally take into consideration the "context" within which the users interact with them [5].

When recommending *motion pictures* in particular, most recommendation systems are based on collaborative filtering. Few content-based movie recommendation systems have been recently proposed, but even those are not based on the content itself, but rather on *metadata* such as directors, actors, genre etc. and on user-provided tags. One of the most sophisticated such systems is *jinni*[1] which uses *semantic tags* to annotate movies instead of flat keywords. This allows it to group tags into categories such as *plot*, *mood*, *location*, etc. and provide more refined recommendations and browsing. Even so, content metadata is still based on a pre-determined tags taxonomy and is produced manually. The resulting annotations are high-level categorizations that can be easily recognized by users, such as "master villain," "good versus evil" for *plot*; "exciting," "stylized" for *mood* and so on. In other words, even the most sophisticated content-based movie recommendation systems *do not take into consideration the content itself* but rely instead on manual annotations about the thematic and affective impression that the content has on users.

In this paper we pursue the far more ambitious goal of making recommendations based on the *multimedia signal extracted from movies*, by automatically inferring annotations regarding photography, camera recording styles, sounds and music, etc. As a first step, in this paper we examine if low-level multimedia features can lead to an

[1] For more details please see http://www.jinni.com

automatically computable notion of *content similarity* that can then be used in the context of a hybrid recommender system or a general movie retrieval framework. Efforts towards this direction are usually limited to particular content and context-dependent limitations, such as emotion, visual-features alone, etc. [6]–[10] Instead, in this paper we present a method that focuses on discovering the way low-level multimodal features correlate to human-provided content similarity judgements without any further qualifications.

In the remainder of this paper we first present the literature on audio-visual and textual feature extraction that forms the background we rely on for low-level feature extraction and similarity estimation (Section II). We then present our data collection methodology for eliciting similarity judgements from human annotators (Section III) and the experimental results from comparing these judgements against automatically computed similarities (Section IV). We close by drawing conclusions and outlining future research (Section V).

II. Feature Extraction and Similaruty Estimation

In this section we present the literature on which we base our similarity metrics as well as the extraction of the features these metrics use. We start with the extraction of two feature vector sequences extracted from the audio and the visual channel and then condensed into one audio and one visual characterization of the video as a whole. It should be notes that the audio characterization pertains to the acoustics of the movie and *not* the spoken content. We then proceed with the extraction of features from the subtitle text, which is used to characterize the movie's spoken content.

A. Audio and visual similarity

A common audio analysis methodology is a two-step *short term* and *mid-term* analysis. In the first step the audio signal is divided into short-term, non-overlapping *frames*; and a feature vector is extracted from each frame. In the second step this sequence of feature vectors is divided into mid-term *segments*. For each feature, the values of all the frames in the segment are aggregates into *feature statistics* that characterizes the segment as a whole.

In the work described here, we divide the audio signal into 40-msec non-overlapping short-term frames and extract the following features from each frame:

- Energy
- Zero Crossing Rate
- Energy Entropy
- Spectral centroid, spread, entropy, flux, and roll-off
- Mel-Frequency Cepstrum Coefficients (MFCCs)

It total, 21 features are extracted. These are standard features and statistics and their precise definitions can be easily retrieved in the audio analysis literature [11]–[13]. We then segment into 2-second mid-term windows with 75% overlap. For each feature, we compute the *average value* and the *standard deviation* of the values in the segment's frames, resulting in a 42-dimensional audio feature space.

The visual channel may contain important information regarding the type of movies and particular movie scenes. In this work, we have focused on extracting basic low-level visual features. In particular, using a 0.5 second of processing analysis (i.e. 2 frames per second), we extract the following features:

- RGB-based histograms. Color distributions can carry very useful information that may correlate with the viewer high level scemantic movie characteristics. In this work, we extract 5 histogram bin from a measure of colorfulness of each frame. This measure is extracted as an average ratio of the maximum RGB value by the gray (average) value.
- Intensity histogram. 8 histogram bins are used to model the intensity values distribution.
- Average intensity difference. This is a single value that corresponds to the average difference on the intensity values of two succesive frames.
- Face information: towards this end the Viola-Jones face detection method [14] has been used to detect faces. The number of faces per frame along with the respective relative (to the overall frame size) bounding box size have been used as features. Thus, two face-relevant features are extracted in total.
- Camera motion information. Towards this end, optical flow [15] has been extracted using the OpenCV implementation of a sparse iterative version of the Lucas-Kanade optical flow in pyramids [16]. Then, these vectors were used to detect horizontal and vertical camera panning moves.

In total, for each 0.5 seconds of visual information, an average feature vector of 17 dimensions is extracted, leading to a final feature matrix of $2 \cdots T_s$, where T_c is the rounded number of seconds of the video's length. Note that this final feature resolution is different to the mid-term time resolution adopted in the audio module. This is not a problem, since no time alignment is needed between the two feature sequences, as time-independent representations are used to extract similarities (see next paragraph). For the particular time resolutions of the two modalities have been selected in order to be able to follow the respective content changes usually occuring in the respective modalities.

After extracting feature vector sequences for the audio and visual modalities, a similarity measure is computed. However, given the time-dependent representations of the audio and the visual modalities, we need a methodology for computing the similarity of two feature vector sequences as a whole, regardless of whether similar segments temporally overlap or not. That is to say, if two movies have similar audio segments, these segments should contribute towards the overall movie similarity even if their their length and location in the overall stream varies.

The standard way to achieve this is to apply *Gaussian Mixture Models (GMM)* to represent each feature set and then apply a simple distance calculations between the extracted mixtures. In particular each audio and video feature set was modeled by a three-component GMM. As a distance measure for GMMs, Euclidean distance has been successfully used in the context of a music information retrieval system [17] where a duration-independent distance measure between music signals was extracted. The two corresponding similarity matrices of audio and visual information were formed using these distances.

B. Textual similarity

In information retrieval and text mining, it is commonly assumed that words (or often longer units, such as n-grams or words) appear independently and that their order is immaterial for the purposes of measuring the similarity in the terminology used in two documents, and thus the thematic similarity between the documents. This leads to *bag of words* representations where the document is represented by a multi-dimensional vector. The vector space is created by assigning a dimension to each word in the document collection and each document is represented by a vector where the value in each dimension reflects the prominense of the corresponding word in the document.

The most straight-forward way is to simply use *term frequency* $\mathrm{tf}(d, t)$, the number of times term t appears in document d. However, we need to consider that the most frequent terms are not necessarily the most informative ones. Even if *stopwords* have been removed in a pre-processing step, there will still be substantive words that are overall frequent in the collection and are not characteristic of a document despite their high frequency. To overcome this, the *term frequency-inverse document frequency (tf-idf)* [18], [19] metric weights the frequency of a term with a document frequency factor that accounts for words that are overall frequent in the collection. More specifically:

$$\mathrm{tfidf}(d, t) = \mathrm{tf}(d, t) \cdot \log \frac{|D|}{\mathrm{df}_D\,(t)}$$

where *document frequency* $\mathrm{df}_D(t)$ is the number of the documents from collection D in which term t appears. *Term frequency* $\mathrm{tf}(d, t)$ is usually defined as the absolute frequency of term t in document $d \in D$. Scaling is sometimes used in size-unbalanced document collections if it is desired to have thematically similary but size-wise unbalanced documents be represented by similar values in their tf-idf vectors.

Besides stopword removal mentioned above, *stemming* is also typically applied in order to unify variations of the same term due to inflectional morphology. For English (and most European languages), the *Porter algorithm* [20] is used to strip inflection suffixes. Stopword extraction, stemming, and similar text pre-processing functionalities are readily provided for many languages and by various text processing frameworks, including the Natural Language Toolkit [21] used in the work described here.

More sophisticated methods offer robustness to *synonymy*, *polysemy* and similar situations where similarity in the words used does not perfectly align with similarity in the meaning. Such methods rely on the observation that the context of ambiguous words determines their semantics and employ *principal component analysis (PCA)* in order to reduce the dimensionality of the representation space. In the new space, roughly speaking, dimensions corresponds to distinct concepts regardless of the word or words used to express them.

In the work described here we use *Latent Semantic Indexing (LSI)* [22] and, more specifically, its implementation in the Gensim library [23]. The main advantage of LSI is that it uses *singular value decomposition (SVD)* in order to reduce the dimensionality of the representation. The mathematics of SVD is such that is preserves the sparsity of the matrices involved; given that we operate in a very large number of dimensions (one for each term encountered in the overall collection), alternative analysis methods that do not preserve sparsity are computationally infeasible.

As soon as documents are represented as feature vectors, we can define similarity between documents to correspond to the similarity between their respective vectors. A popular similarity measure in information retrieval is *cosine similarity*, the cosine of the angle between the two vectors:

$$\mathrm{CosSim}(a, b) = \frac{\vec{t_a} \times \vec{t_b}}{|\vec{t_a}| \times |\vec{t_b}|} \tag{1}$$

where $\vec{t_a}$ and $\vec{t_b}$ are the vector representations of documents a and b respectively.

III. Dataset

A. Data description

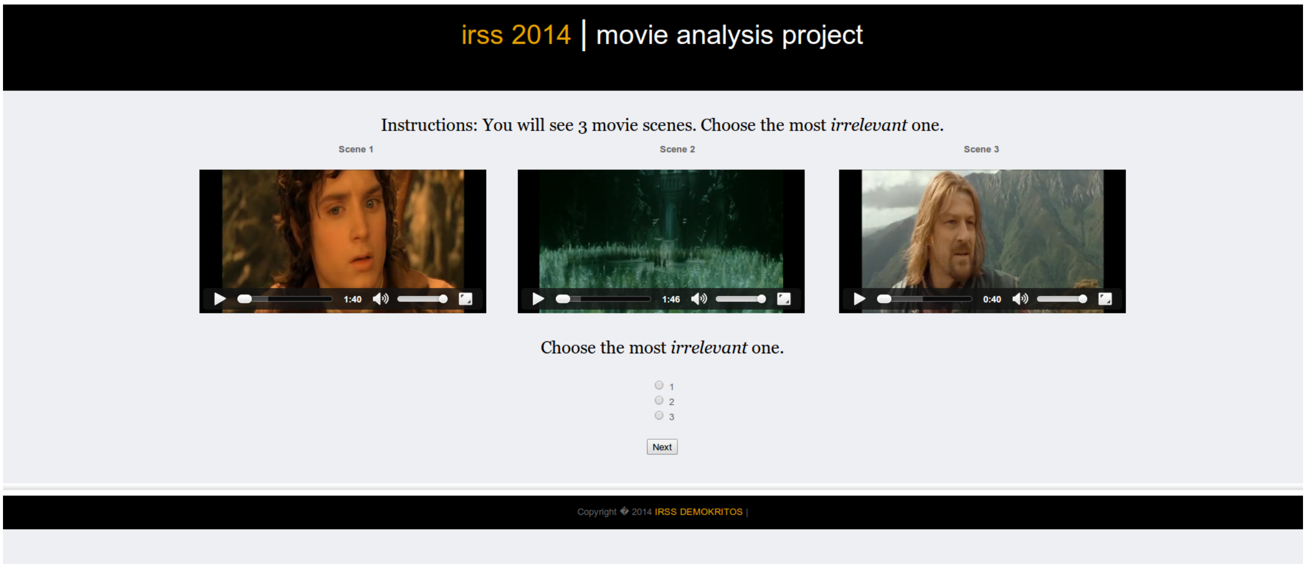
In order to demonstrate the correlation between low-level multimedia features and content similarities, we have manually compiled and annotated a small dataset of movie scenes from the *Lord Of The Rings* trilogy. To this end, we manually selected what we considered the 9 most characteristic scenes from each movie in the trilogy, leading to a dataset of 27 short scenes.

We have chosen to generate a dataset from these three closely related movies in order to avoid metadata-specific bias (genre, casting, location, etc) when annotating similarities between scenes. Instead, the focus of our work is content-derived similarity between scenes.

The scenes were selected for being *homogeneous,* so that the whole scene can be similar or not to another scene. To be homogeneous they are relatively short, with a runtime ranging from 30 sec to 5 min and an average of 2.4 min. They are also selected to be *characteristic,* so that the set of 9 scenes completely characterizes the movies in the sense that they capture all the major ways the movie as a whole can be similar to another movie.

B. Annotation and ground-truth generation

In order to evaluate the proposed similarity extraction techniques we need a ground-truth similarity between the video scenes of the dataset. Towards this end, a web tool has been developed (see Figure 1) in order



irss 2014 | movie analysis project

Instructions: You will see 3 movie scenes. Choose the most *irrelevant* one.

Choose the most *irrelevant* one.

○ 1
○ 2
○ 3

Copyright � 2014 IRSS DEMOKRITOS |

Figure 1: Screenshot of the annotation web tool. The user chooses the most “irrelevant” video

to provide a user friendly interface for humans to make manual similarity annotation of scenes. In this tool, after a simple login page users are presented with three randomly selected scenes and are asked to choose the one that does not “fit” the other two. This is repeated for as long as the annotators want.

The result of this process is a separate similarity matrix $U(u)$ for each annotator u constructed as follows: for each pair of scenes i, j, $U_{i,j}(u)$ is the number of times that u was presented with a scene triplet comprising i, j and some other scene and chose the other scene as the odd one out of the three; minus the number of times that one of i, j was chosen as the odd one out. Positive values correspond to similarity, while negative values correspond to dissimilarity.

This setup was preferred of one where we simply ask annotators to rate or assert/refute similarity between two scenes in order to avoid having to explain what “similarity” means. By presenting similarity judgements as a “pick the odd one out” game, we can avoid explanations that might impose our own perception of “similarity” and instead let the annotators apply their own, unbiased intuitions.

Only the judgements from the 44 annotators that have stayed on for at least 6 scene triplets were retained in the dataset. These 44 annotators provided a total of 5000 pair-wise values for U, out of the $44 \cdot 27 \cdot (27-1)/2 = 15444$ positions in U.[2] In order to generate the ground-truth similarity matrix from all users, we assume the average over all judgements across all annotators, for those pairs where 3 or more judgements have been made. That is, if $N_{i,j}$ is the number of judgements for pair i, j, then the similarity matrix is:

$$SM_{i,j} = \begin{cases} \sum_u U_{i,j}(u)/N_{i,j} & \text{if } N_{i,j} \geq 3 \\ \\ \text{undefined} & \text{otherwise} \end{cases}$$

Furthermore, we define the “agreement matrix” $A_{i,j}$ containing the fraction of annotators that agree regarding pair i, j, i.e., have the same polarity in $U_{i,j}(u)$. Specifically:

$$A_{i,j} = \frac{\max\{Pos_{i,j}, Neg_{i,j}\}}{Pos_{i,j} + Neg_{i,j}}$$

where $Pos_{i,j}$ is the number of annotators for which $U_{i,j}(u) > 0$ and $Neg_{i,j}$ is the number of annotators for which $U_{i,j}(u) < 0$. The average value of the non-diagonal elements of this matrix is the *average inter-annotator agreement*.

Using the 5000 judgements in our dataset, 340 (97%) out of the $27 \cdot (27-1)/2 = 351$ positions in $SM_{i,j}$ are defined with an average inter-annotator agreement of 77%.

C. *User clustering*

The purpose of this step is to evaluate the extracted similarities against groups of similar users. Towards this end, we have extracted “clusters of users” by firstly calculating a user similarity matrix:

$$USM_{u,v} = \frac{M_{u,v}}{N_{u,v}}$$

where $N_{u,v}$ is the number of common annotations and $M_{u,v}$ is the number of common annotations with agreement. As a second step, we adopt a hierarchical clustering approach [12], [24] to discover groups of similar users.

IV. Experiments

A. *Experimental Setup*

Figure 2 presents the overall sceheme of the methodology followed to extract the movie similarities. In general, the following steps are carried out:

[2] All triples have three *different* scenes, so that the diagonal is not computed; furthermore, the ordering inside a triplet is not significant, thus $U_{i,j}(u) = U_{j,i}(u)$.

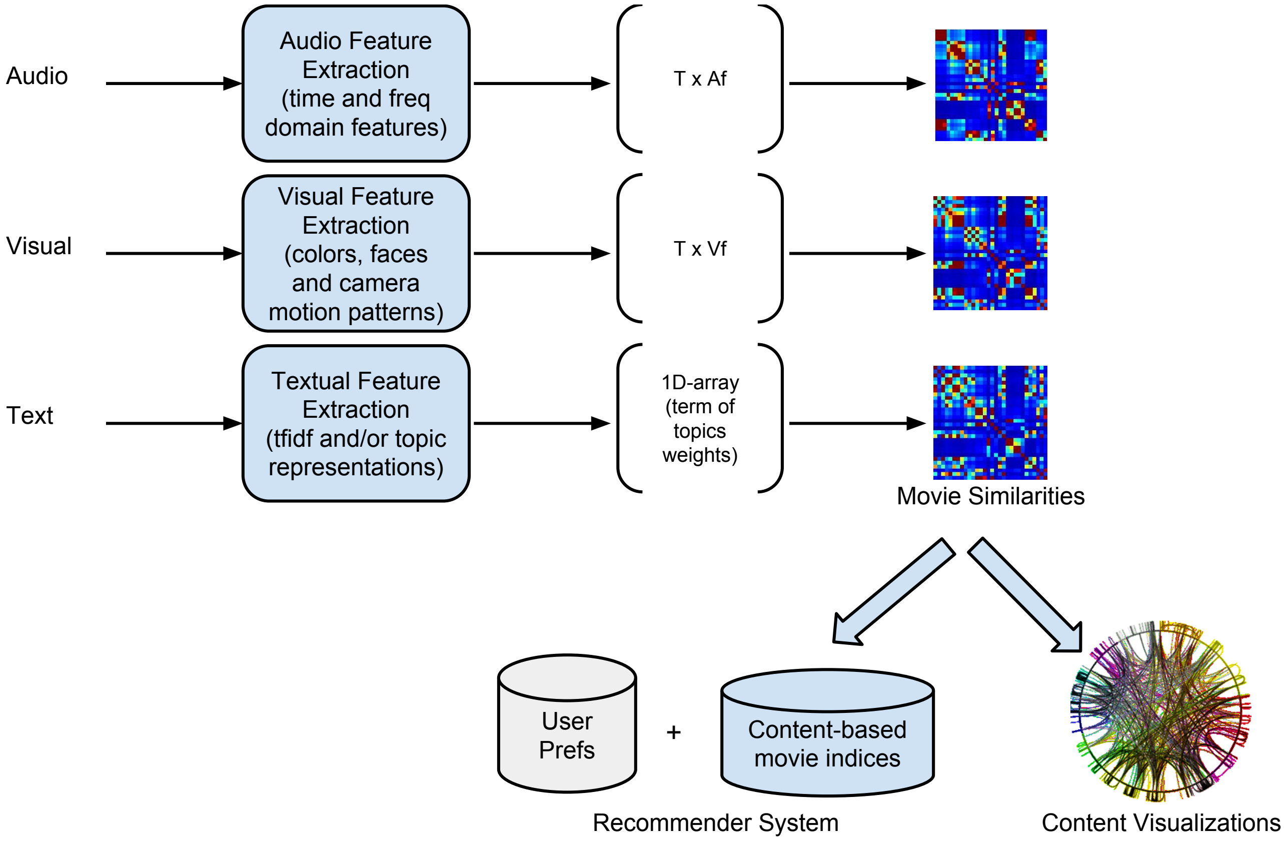

Audio
Audio Feature Extraction (time and freq domain features)
T x Af
Visual
Visual Feature Extraction (colors, faces and camera motion patterns)
T x Vf
Text
Textual Feature Extraction (tfidf and/or topic representations)
1D-array (term of topics weights)
Movie Similarities
User Prefs
+
Content-based movie indices
Recommender System
Content Visualizations

Figure 2: General diagram of the proposed method

- Audio analysis: common short and mid-term audio features are extracted for each audio sequence of a movie. This leads to a sequence of audio feature vectors. The length of that sequence is obviously dependent on the movie duration.
- Visual analysis: color, motion and face-related features are extracted in order to model the visual information. Similar to the audio domain, this yelds to a time sequence of feature vectors.
- Subtitle analysis: a topic modelling approach is followed in order to represent each set of subtitles.

We have adopted two simple early fusion approaches in order to combine the similarities obtained from the individual modalities:

- Simple average; and
- Weighted average, where weights are obtained from the overall accuracy of each individual modality.

Such a content-specific similarity can be used in a hybrid (collaborative and content-based) recommended system. In addition, as shown in the experimental evaluation, such a similarity indexing should be user-specific, as we observed above user clusters that indicate that different humans have different perceptions of what is similar.

B. Experimental Results

In order to compare the similarity matrices with the ground-truth matrix it is necessary to define a set of performance measures. In particular, we followed an information retrieval driven-rationale: for each row of the ground truth similarity matrix (i.e. for each video of the dataset) we return the most similar videos. The number of the most returned (similar) scenes changes per row and is calculated as follows:

$$N_i = |\{j, \text{ such that } SM_{i,j} > T\}|$$

where $T = C \cdot \mu_i$, μ_i the average value of the i-th row of the similarity matrix and C a user defined parameter (set equal to 1 for our case). In practice, N_i is the number of elements of the i-th row that are more similar than a threshold that depends on the average values of similarities in that row. The exact same procedure is executed for the automatically extracted similarities, from the text, visual and audio domains, leading to a different set of "most similar" videos for each video in the dataset.

Based on these two sets of returned videos (ground truth and automatically retrieved), we adapt the concepts of *precision* and *recall* as follows: *Precision* is the number of correctly returned results (i.e. the number of results that also belong to the ground truth set for the respective scene) divided by the total number of relevant scenes (according to the respective row of the ground truth similarity matrix). *Recall* for a particular row (video scene) is the number of correctly returned results (i.e. the number of results that also belong to the ground truth set for the respective scene) divided by the total number of returned results (by any of the automatically extracted results).

It should be noted that precision and recall are computed row-wise, that is, for each video (scene) and are *averaged*

Table I: Overall performance measures (%) for each modality and both fusion approaches

Method	Rec	Pre	F1
Text	40.5	25.0	31.1
Audio	67.2	20.5	31.3
Visual	56.2	22.8	32.4
Fusion 1	59.1	23.3	33.5
Fusion 2	58.0	24.5	34.0

Table II: Overall performance measures (%) for each modality, compared to the ground-truth that corresponds to clusters of users

Method-Cluster	Rec	Pre	F1
CL1-Text	41.9	26.4	32.4
CL1-Audio	72.9	21.2	**32.8**
CL1-Visual	54.4	20.4	29.7
CL2-Text	32.4	24.9	28.1
CL2-Audio	65.2	22.8	33.8
CL2-Visual	57.2	26.3	**36.0**
CL3-Text	40.9	25.3	31.3
CL3-Audio	69.2	20.9	**32.1**
CL3-Visual	53.8	21.7	30.9

into the overall precision and recall of the experiment. The overall *F1 score* is computed from these averages.

Table I presents the performance measures on the whole dataset. It can be seen that the fusion boosts the performance related to each individual classifier and that the best individual modality is audio and visual, however just slightly better than text. Finally, we note that the random selection retrieval achieves a baseline performance of 17%.

Table II presents the evaluation results compared to the ground-truth as extracted by groups of similar users. Note that for some clusters of users, particular modalities outperform compared to the average performance of Table II. This indicates that some group of users are preferable to one particular type of modality and their decisions regarding the similarity between scenes they were mostly based on one type of low-level features. In other words, there are clusters of users that correlate between each other in particular modalities. This indicates that the way the users correlate to low-level features is not uniform for all modalities: some users judge the content similarity based mostly on audio, others based on visual information, etc. This is a rather important observation since it can provide a useful tool in a content-based recommendation system that personalizes retrieved results according to particular low-level features from particular media.

V. Conclusion

The core idea of this work was not to create a full content-based recommendation system, which is actually a much larger-scale project, but to examine the correlation between low-level audio, visual and textual features and movie content similarity. Detailed experimental evaluation has led to the conclusion that these features correlate with human perception of what is similar, in terms of content. In addition, it has been demonstrated that two simple methods for fusion of the modalities outperforms (in terms of retrieval accuracy) each individual modality. Finally, it has been shown that different modalities correlate better for different clusters of users, giving us an background knowledge to consider a new way of combining content-based recommendation approaches with personalization functionalities.

The system and the experimental results described above have yielded several interesting observations and useful outcomes:

1) a similarity estimation methodology that collects and integrates established feature extraction and feature vector comparison method from the literature, as well as a methodology for collecting human judgements to evaluate these similarity estimates;
2) the empirical validation of our core premise that there is a correlation between low-level features that can be automatically extracted from movies and human similarity judgements;
3) the observation that even a very simple *modality fusion* boosts the performance of the retrieval accuracy, which indicates that there is a certain amount of diversity in the individual media.
4) the observation that that the level of correlation between the low-level features and clusters of humans (based on what they interpret as similar content) is *not uniform*: some clusters of humans correlate better with the textual audio features, others with the audio-visual. This is a very important finding that can be used in the context of a content-based recommendation system that personalizes retrieved results according to particular low-level features from particular media.

It should be noted that all code and manual annotations are publicly available on Github.[3]

These findings helped us design a research path towards content-based similarity measures that can be used in the context of a large-scale recommendation system:

- experimenting with state-of-the-art clustering approaches and with more information regarding user preferences and previous ratings, aiming to *predict* user categories, instead of relying on the similarity annotations themselves to classify users;
- work towards more sophisticated modality fusion, and in particular dynamically using user clustering results in order to define a *personalized* modality fusion method;
- experimenting with fusing content similarity with *movie metadata* (cast, director, etc.) and *user preferences* (collaborative information) and comparing the predictive power of the different types of features;
- developing methodologies for automatically segmenting movies into scenes. In the work described here we have manually carried out segmentation based on our own understanding of how these scenes will be used; we need to proceed by rigorously defining what *homogeneous* and *characteristic* scenes means

[3] Please see https://github.com/lehinevych/irss2014-movie and see README.md for an explanation of the data and code in the repository.

in automatically extractible terms and by experimenting with the impact of segmentation errors on the similarity estimates between whole movies.

At a more technical level, we will need to implement feature extraction and similarity computation techniques efficiently and scalably. We are also planning to develop interactive functionalities using advanced *visualization* techniques for browsing movies and interactively customizing recommendations.

Acknowledgment

The work described here was carried out at IRSS 2014, the 2014 International Research-Centred Summer School (http://irss.iit.demokritos.gr)

We are most grateful to fellow IRSS students, colleagues at NCSR, and members of the general public for donating their time and effort to provide similarity annotations.

References

[1] F. Ricci, L. Rokach, B. Shapira, and P. Kantor, Eds., *Recommender Systems Handbook*. Springer, 2011.

[2] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender systems: an introduction*. Cambridge University Press, 2010.

[3] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The Adaptive Web*. Springer, 2007, pp. 325–341.

[4] L. M. De Campos, J. M. Fernández-Luna, J. F. Huete, and M. A. Rueda-Morales, "Combining content-based and collaborative recommendations: A hybrid approach based on bayesian networks," *International Journal of Approximate Reasoning*, vol. 51, no. 7, pp. 785–799, 2010.

[5] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender systems handbook*. Springer, 2011, pp. 217–253.

[6] L. Canini, S. Benini, and R. Leonardi, "Affective recommendation of movies based on selected connotative features," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 23, no. 4, pp. 636–647, 2013.

[7] Z. Rasheed, Y. Sheikh, and M. Shah, "On the use of computable features for film classification," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 15, no. 1, pp. 52–64, 2005.

[8] M. Szomszor, C. Cattuto, H. Alani, K. O'Hara, A. Baldassarri, V. Loreto, and V. D. Servedio, "Folksonomies, the Semantic Web, and movie recommendation," in *Proceedings of the 4th European Semantic Web Conference (ESWC 2007), Innsbruck, Austria, June 2007*, 2007.

[9] O. Kirmemis and A. Birturk, "A content-based user model generation and optimization approach for movie recommendation," in *Workshop on ITWP*, 2008.

[10] A. Zenebe and A. F. Norcio, "Representation, similarity measures and aggregation methods using fuzzy sets for content-based recommender systems," *Fuzzy Sets and Systems*, vol. 160, no. 1, pp. 76–94, 2009.

[11] T. Giannakopoulos and A. Pikrakis, *Introduction to Audio Analysis: A MATLAB Approach*. Academic Press, 2014.

[12] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, Fourth Edition*. Academic Press, Inc., 2008.

[13] K. Hyoung-Gook, M. Nicolas, and T. Sikora, *MPEG-7 Audio and Beyond: Audio Content Indexing and Retrieval*. John Wiley & Sons, 2005.

[14] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[15] B. K. Horn and B. G. Schunck, "Determining optical flow," in *1981 Technical Symposium East*. International Society for Optics and Photonics, 1981, pp. 319–331.

[16] J.-Y. Bouguet, "Pyramidal implementation of the affine Lucas Kanade feature tracker – description of the algorithm," *Intel Corporation*, vol. 5, 2001.

[17] M. Helen and T. Virtanen, "Query by example of audio signals using euclidean distance between gaussian mixture models," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 1, April 2007, pp. I–225–I–228.

[18] R. Baeza-Yates, B. Ribeiro-Neto *et al.*, *Modern information retrieval*. New York: ACM Press, 1999, vol. 463.

[19] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge: Cambridge University Press, 2008, vol. 1.

[20] M. F. Porter, "An algorithm for suffix stripping," *Program: electronic library and information systems*, vol. 14, no. 3, pp. 130–137, 1980.

[21] S. Bird, "NLTK: The Natural Language Toolkit," in *COLING/ACL on Interactive presentation sessions*. Association for Computational Linguistics, 2006, pp. 69–72.

[22] S. Dumais, G. Furnas, T. Landauer, S. Deerwester, S. Deerwester *et al.*, "Latent semantic indexing," in *Proceedings of the Text Retrieval Conference*, 1995.

[23] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.

[24] T. Warren Liao, "Clustering of time series data–a survey," *Pattern recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.