Enhanced Semantic TV-Show Representation for Personalized Electronic Program Guides

Cataldo Musto[1], Fedelucio Narducci[1], Pasquale Lops[1], Giovanni Semeraro[1], Marco de Gemmis[1], Mauro Barbieri[2], Jan Korst[2], Verus Pronk[2], and Ramon Clout[2]

[1] Department of Computer Science, University of Bari "A. Moro", Italy
{cataldomusto,narducci,lops,semeraro,degemmis}@di.uniba.it
[2] Philips Research, Eindhoven, The Netherlands
{mauro.barbieri,jan.korst,verus.pronk,ramon.clout}@philips.com

Abstract. Personalized electronic program guides help users overcome information overload in the TV and video domain by exploiting recommender systems that automatically compile lists of novel and diverse video assets, based on implicitly or explicitly defined user preferences. In this context, we assume that user preferences can be specified by *program genres* (documentary, sports, ...) and that an asset can be labeled by one or more program genres, thus allowing an initial and coarse preselection of potentially interesting assets. As these assets may come from various sources, program genre labels may not be consistent among these sources, or not even be given at all, while we assume that each asset has a possibly short textual description. In this paper, we tackle this problem by considering whether those textual descriptions can be effectively used to automatically retrieve the most related TV shows for a specific program genre. More specifically, we compare a statistical approach called *logistic regression* with an enhanced version of the commonly used vector space model, called *random indexing*, where the latter is extended by means of a negation operator based on quantum logic. We also apply a new feature generation technique based on *explicit semantic analysis* for enriching the textual description associated to a TV show with additional features extracted from Wikipedia.

Keywords: Personalized Electronic Program Guides, Explicit Semantic Analysis, Vector Space Model, Random Indexing, Logistic Regression.

1 Introduction

The world of television has changed dramatically in the last few years. People used to have access to a few tens of television channels. Then, with the advent of digital satellite receivers, these few tens channels became a few hundred channels. More recently, the number of channels has become practically unlimited if we count the billions of videos that websites such as YouTube offer. We have never seen so many options for finding and accessing videos. While having some options is more desirable than having no choice, it is known that having too many choices

J. Masthoff et al. (Eds.): UMAP 2012, LNCS 7379, pp. 188–199, 2012.

leads eventually to dissatisfaction [12]. One possible solution to this overload problem is represented by *personalized electronic program guides* (EPGs).

Personalized EPGs help users find relevant (TV and Web) video content by using recommender systems. A possible approach for realizing a personalized EPG [19] is to divide the task into two steps. A first step consisting of a coarse filtering of the available assets in predefined categories, followed by a ranking step based on the application of a recommender system employing a learned user profile specific for each category. For the first filtering step, it is common practice to classify TV shows by labeling them with one or more *program genre* labels, such as *documentary*, *sports*, etc.

This two-step approach also helps overcoming scalability issues that arise if we want to consider the suitability of each individual asset for each individual user. More specifically, if N is the number of available videos and M is the number of users, we want to avoid that we have to consider each of the $N \cdot M$ combinations in looking for matches.

As digital video asset originate from various sources (e.g. YouTube, broadcast TV, video-on-demand libraries) we may not expect that assets are consistently labelled with one or more program genres. While some sources may not even provide labels, we do expect that each video asset has associated a title and a possibly short textual description.

We believe it makes sense to use program genres as intermediate specification medium to make a first, coarse preselection of potentially interesting assets. In this paper we focus on the problem of automatically mapping the textual descriptions of video assets to program genres. We compare two machine learning methods used to compile a ranked list of TV shows for each program genre. We also investigate how to enrich short textual descriptions with more informative keywords using knowledge automatically extracted from Wikipedia. Our experimental results are obtained on a large collection of TV-show descriptions.

2 Motivating Scenario

This research is carried out in the context of APRICO Solutions, a software company that is part of Philips Electronics (see www.aprico.tv), which develops video recommender and targeting technology, primarily for the broadcast and Internet industries. The EPG data used in this research is provided by Axel Springer (see www.axelspringer.de), a strategic partner of APRICO Solutions.

One of the concepts developed at APRICO Solutions is the concept of *personal channels*. A user can create a personal channel by selecting a TV show or an Internet video asset as seed. Based on the seed attributes, similar TV shows and Internet videos are automatically selected and aggregated into a playlist that can be viewed as a linear channel next to the traditional broadcast TV channels. The order of the videos in the playlist of a channel is typically based on time of broadcast or relevance of the content to the channel. Users can add and delete programs from the playlist at will. The basic architecture of a personal channel is shown in Figure 1. Each channel has a boolean filter that preselects TV shows

and Internet videos based on the characteristics of the video seed used to create the channel. The shows that pass the filter are prioritized by a recommender that learns from the interaction of the user with the channel and through explicit ratings. Note that in this concept, users are not explicitly modeled, but their multiple interests and preferences are captured by the multiple personal channels, each having a dedicated recommender.

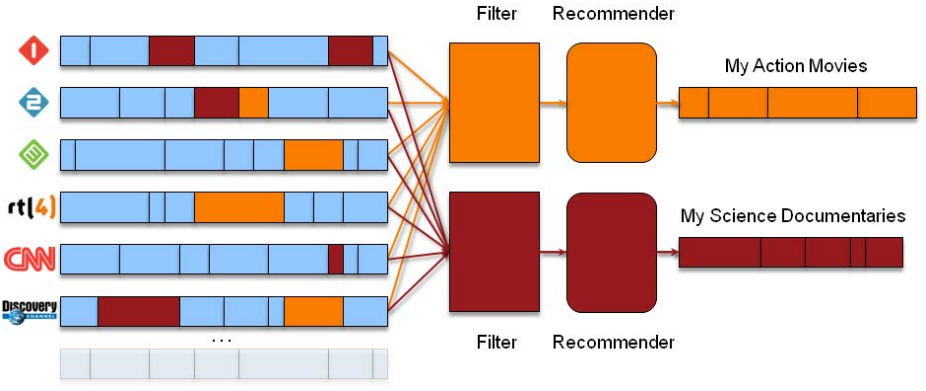

Fig. 1. The personal channel concept

An important attribute used by the boolean filters to preselect shows for a personal channel is the *program genre*. Examples of program genres are *movie*, *sports*, *documentary*, and *TV series*. Given that many video assets do not have associated an explicit program genre (e.g. videos from Internet video portals), the problem addressed in this paper is: given a program genre, automatically retrieve a ranked list of TV shows and Internet videos that match the given program genre. More formally, we define the problem as follows:
TV-show ranked retrieval task. Given $S = \{s_1, \ldots, s_n\}$ a set of TV-show descriptions, and given a program genre $p \in P = \{p_1, \ldots, p_m\}$, where P is a set of program genres, return a ranked list of k TV-show descriptions from S that best match program genre p.

3 TV-Show Representation Using Explicit Semantic Analysis

A simple and convenient way to represent textual descriptions of TV shows is called *bag of words* (BOW), in which each item is represented by the set of words in the text, together with their number of occurrences. In this work we compare the classical BOW representation with an enhanced one (E-BOW) built by enriching the classical BOW model with additional features automatically extracted from Wikipedia. To this purpose we exploited a technique called *explicit*

semantic analysis (ESA) [11] that allows to represent terms and documents using Wikipedia pages (concepts). In order to describe how ESA works, we assume that each article in Wikipedia is a concept. Given a set of concepts $C_1, C_2, ..., C_n$ and a set of associated documents $d_1, d_2..., d_n$ (the Wikipedia articles themselves), we construct a sparse matrix T, called ESA-*matrix*, where each of the n columns corresponds to a concept, and each row corresponds to a term (word). The cell $T[i, j]$ of the matrix represents the TFIDF value of term t_i in document d_j.

After building the ESA-*matrix*, for each term we are able to extract its *semantic interpretation vector* that is the corresponding row in the ESA-*matrix*. A *semantic interpretation vector* for a text fragment (i.e. a sentence, a BOW, an entire document) is obtained by computing the centroid (average vector) of all the *semantic interpretation vectors* related to the terms occurring in that specific text fragment.

ESA is exploited to generate the E-BOW by adding new features to the original TV-show textual descriptions. As the TV-show descriptions in our dataset are in German, we processed the German Wikipedia dump released on October 13th, 2010 (approximately 7.5 GB). After the processing step and the application of the heuristics described in [11] in order to narrow the number of terms and Wikipedia articles in the ESA-*matrix*, we obtained a matrix with 814,013 rows (terms) and 484,218 columns (Wikipedia articles). The input of the feature generation step is the whole BOW considered as a unique text fragment. We adopt this strategy because TV-show descriptions are quite short, and it is difficult to split the text in several fragments (i.e sentence, paragraph, etc.). For each term in the BOW, the corresponding vector from the ESA-*matrix* is extracted, and the centroid of all those vectors is computed. The final step consists in selecting the most important Wikipedia concepts from the centroid vector (those with a higher weight) for adding to the original BOW. The new E-BOW is composed by the keywords in the BOW and the new generated features (Wikipedia concepts) extracted by the centroid vector.

Figure 2 provides an example of a set of features generated for a TV show belonging to the *sports* program genre titled *Rad an Rad - Die besten Duelle der MotoGP (Wheel to wheel - The best duels in the MotoGP)*. We can observe that new concepts related to MotoGP motorcyclists (*Valentino Rossi, Max Biaggi, Shin'ya Nakano, Loris Capirossi*), MotoGP competitions (*großer preis von italien - Italian motorcycle Grand Prix, großer preis von malaysia - Malaysia motorclycle Grand Prix*, etc.), and other generic concepts such as *motogp* have been introduced. Hence, the idea behind the feature generation process is to introduce new concepts allowing an easier identification of the right program genre for a specific TV show. In a recommendation scenario, that representation has several advantages. First of all, representing user interests in terms of (comprehensible) Wikipedia articles allows obtaining a more *transparent* user profile. Furthermore, *serendipitous* (unexpected) recommendations may also be produced: in the previous example the Wikipedia concept *scuderia ferrari* is not directly related to the analyzed TV show (see Figure 2), but it might be interesting for the user.

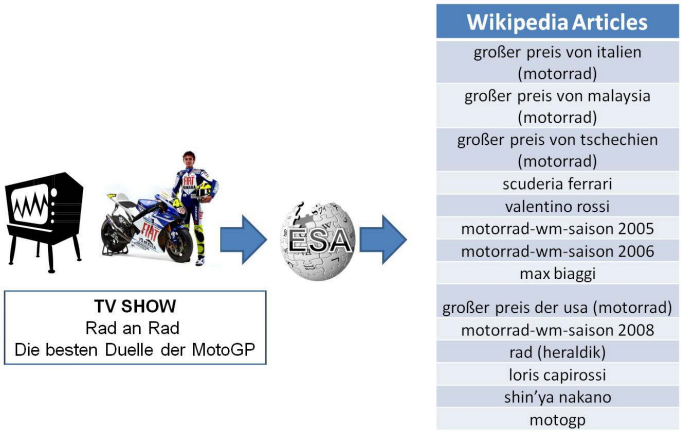

Fig. 2. An example of enrichment by ESA

4 TV-Show Ranked Retrieval

We investigate the application of two different machine learning approaches for the task of TV-show ranked retrieval. The first approach is *random indexing* (RI), a strategy which has been shown to be more effective than the classical *vector space model* (VSM) [17] in a recommendation scenario, while the second approach is *logistic regression* (LR), a gold standard in text classification, which has been adapted for the ranked retrieval task.

Random Indexing. RI is an efficient, scalable and incremental technique for dimensionality reduction. It belongs to the class of so-called *distributional models*, which state that the meaning of a word can be inferred by analyzing its use (that is to say, its *distribution*) within a corpus of textual data. By following this approach, we can represent terms and documents as points in a vector space with a considerable reduction of features to describe them. Through this model we can obtain results comparable to other well-known dimensionality reduction methods (such as singular value decomposition, applied in LSA [7]), but with a substantial saving of computational resources. The goal of RI is to shift the classical VSM representation based on a n-dimensional term-document matrix towards a k-dimensional term-context matrix that is more compact and flexible, since the number of contexts (i.e. the *dimension of the matrix*) is not fixed and could be adapted to the requirements of the specific application domain. The context of a term could be a sliding window of a couple of terms that surround it on the left and on the right, a whole sentence, a paragraph, or the whole document. In this work we exploit the simplest formulation we can provide, namely the context of a term is defined as the *whole document* in which it occurs.

The first step consists of reducing the vector space through the RI algorithm. As in the Rocchio classification algorithm [21], a prototype vector is built for each program genre by summing up all the TV-show vectors belonging to that

program genre. Given a prototype vector, we compute the cosine similarity with all TV shows to get the list of the best matching TV-show descriptions for a specific program genre. RI has been extended by means of the quantum negation operator in order to model also the negative evidences for program genres, the terms that typically do not occur in descriptions of a given program genre. The negation operator is useful to get the subspace that will contain the items as close as possible to the positive preference (liked program genre) vector and as far as possible from negative one. It has been already shown that the introduction of the negation operator allows obtaining better results [17].

Logistic Regression. LR is a supervised learning method able to analyze data and recognize patterns. It can be applied in cases where the dependent variable we want to predict can have as value 0 or 1 (i.e., a TV show belongs or does not belong to a specific program genre). The learned model represents the examples as points in a multidimensional space and a logistic function is learned for each class. A logistic function is represented by a sigmoid curve. LR is used for text classification tasks, achieving similar results to *support vector machines* (SVM) [26]. LR has a good accuracy, is robust, and fully automatic, eliminating the problem of manually tuning parameters. LR produces probabilities as output and is preferred over SVM in those scenarios where this aspect has a high relevance. In this work we use the LIBLINEAR library [10], an open-source library for large-scale linear classification (for datasets with a huge number of features and instances) that supports LR and SVM. Given a program genre p, the TV shows are ranked based on their probability to belong to p and are returned in a ranked list. It is possible that the same TV show belongs to the retrieved list of different program genres, but with a different probability value, and a different position in the ranked list. For example, if we have a documentary about horses and equestrian disciplines, it could belong to the retrieved list of "documentary" as well as "sports".

5 Experimental Evaluation

We carried out two distinct experimental sessions: the goal of the first one is to measure the effectiveness of RI and LR in the ranked retrieval task, while the goal of the second session is to investigate the effectiveness of the feature generation process. The dataset used in the experiments contained a set of 133,579 TV-show descriptions, from a set of 47 broadcast channels in the German language. TV shows have been broadcast between April 2009 and April 2011. We assumed that one program genre is specified for each TV show. A TV-show description has an average length of approximately 42 word occurrences.

We run the whole experimental evaluation through a *k-fold cross validation*, with k=10. The dataset has been partitioned into 10 subsets of equal size and 10 different runs have been evaluated, each using a different subset for testing and the rest for training.

In Figure 3 the distribution of the TV shows among all the categories is plotted. The dataset is very unbalanced towards some program genres such as

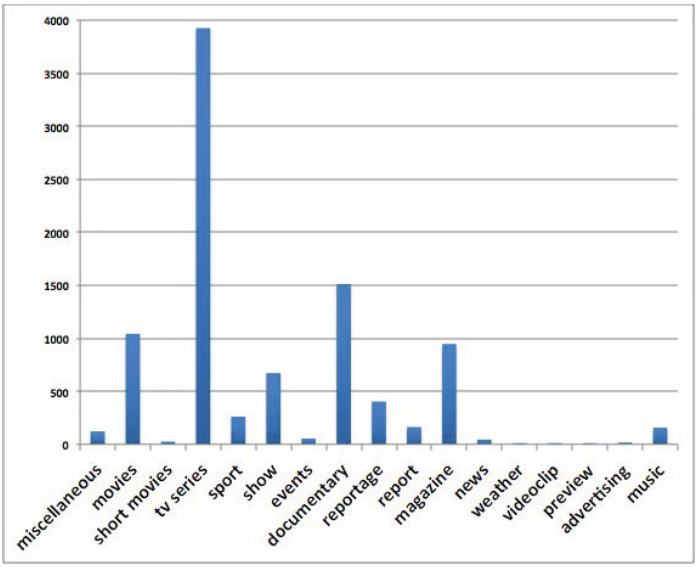

Fig. 3. Distribution of the training examples among the 17 program genres

TV series, *movies* and *documentary* in comparison to other program genres with a very small number of instances, such as Weather.

We used the precision at $n\%$ with $n = \{5, 10, 25, 50, 75, 100\}$, as metric for evaluating the effectiveness of the proposed model. For a given program genre p, let $L(p)$ be the ranked list of TV shows that are retrieved for program genre p. Let $L(p, n)$ be the *top-n* percent of $L(p)$. Note that the length of $L(p)$ is given by the number of items that actually have program genre p in the given test set. Furthermore, let $C(p, n)$ denote the subset of correctly classified descriptions in $L(p, n)$. Then, the precision at $n\%$, denoted by $P@n\%$ is given by:

$$P@n\% = \frac{|C(p,n)|}{|L(p,n)|} \ . \tag{1}$$

5.1 Experiment 1: Evaluation of the Ranked Retrieval Task

The goal of this experiment is to compare the RI and LR performance for the ranked retrieval task. For RI, we evaluated the performance using different sizes of context vectors (the k parameter in Section 4). More specifically, we evaluated the effect of reducing the dimensionality of vectors representing the TV shows. Indeed, the original size of feature vectors representing TV-show descriptions is $133,579$. In our experiment we considerably reduced the size of those vectors to 500, $1,000$, $1,500$ and $2,000$.

Table 1. Results of RI and LR algorithms on the ranked retrieval task ($P@n\%$)

Approach	k	$P@5\%$	$P@10\%$	$P@25\%$	$P@50\%$	$P@75\%$	$P@100\%$
RI	500	0.842	0.791	0.709	0.632	0.578	0.528
RI	1000	0.850	0.802	0.722	0.648	0.591	0.543
RI	1500	0.855	0.806	0.732	0.653	0.599	0.548
RI	2000	0.851	0.810	0.732	0.656	0.600	0.551
LR		**0.920**	**0.903**	**0.884**	**0.864**	**0.820**	**0.747**

Results are depicted in Table 1. It is worth to note that the different size of the context vectors does not affect the performance of the RI algorithm. The most important result is that LR outperforms RI: the larger the number of retrieved items, the larger the gap between the performance of RI and LR. This results confirms that RI is more effective on the retrieving TV shows ranked in the first positions of the list. This is confirmed by observing the loss of performance of RI with an increasing number of retrieved items, as well. On the contrary, LR preserved its accuracy also by considering the whole retrieved list of items ($P@100\%$). We can conclude that LR achieves the best performance, even in the case of large retrieved lists.

5.2 Experiment 2: Evaluation of the Feature Generation Process

In the second experiment we evaluated the impact of the feature generation process described in Section 3 on the performance of the retrieval algorithm. Since LR achieved the best performance in the first evaluation, we decided to run the second evaluation using that algorithm. For each BOW associated to a TV show, we used the ESA-*matrix* to extract the 20, 40 and 60 most related Wikipedia concepts, that we added as new features for enriching the original BOW. Table 2 depicts results of the experiment. It is worth noting that all the configurations using the E-BOW representation outperform the BOW baseline, even though differences between results using 40 or 60 concepts seem to be not statistically significant.

Table 2. Comparison between BOW and enriched BOW ($P@n\%$)

Metric	BOW	E-BOW+20	E-BOW+40	E-BOW+60
$P@5\%$	0.920	0.921	0.941	**0.943**
$P@10\%$	0.903	0.912	0.935	**0.937**
$P@25\%$	0.884	0.902	0.924	**0.927**
$P@50\%$	0.864	0.880	0.901	**0.903**
$P@75\%$	0.820	0.838	0.864	**0.867**
$P@100\%$	0.747	0.764	0.785	**0.786**

Hence, we run the Mann-Whitney Test [20] that, given two sets of observations (obtained by two different approaches) and a single ordering of those results, is able to decide whether the ranked list is achieved by chance or not. We performed the test for each of the following pairs of results by comparing the data points that were obtained in the 10-fold cross validation:

- BOW vs E-BOW+20
- BOW vs E-BOW+40
- BOW vs E-BOW+60

For each level of precision all the results are statistically significant ($p < 0.05$) except the difference between BOW and E-BOW+20 in terms of $Pr@5\%$. We can thus conclude that the ESA-*based* feature generation process allows a better ranking of the most relevant items for each program genre, and this is a very interesting result since in a recommendation scenario a limited number of TV shows is generally suggested.

6 Related Work

The literature in the area of TV recommendation dates back to the 1990s [9]. One of the first attempt in the area of personalized TV that exploits information filtering techniques is presented in [14]. The authors produce a personalized list of TV news according to duration constraint, and solve the problem by an optimization model. Two prototypes were proposed: a category-based system and a keyword-based one. The main differences with respect to our model is that categories are manually assigned to each news, and content was not subjected to any enrichment process. Also in [23], the proposed hybrid personalization techniques produce suggestions according to the program categories a target user has enjoyed in past. In that work a combination of content-based and collaborative recommendation strategies was proposed. However, a problem emerged in that research was the weak diversity of content-based recommendations. The authors proposed a hybrid model for this purpose. To overcome that limitation, in our model we infuse new Wikipedia-based knowledge in the TV-show descriptions.

A collection of research reports on the development of personalized services for Interactive TV is provided in [1], while a thorough overview of current research and trends in the field of personalized TV applications is in [6].

Regarding the dimensionality reduction problem, Berry et al. [4] pointed out the need for dimensionality reduction techniques as a mean to improve the effectiveness and the scalability of VSM. In this context effective techniques for dimensionality reduction such as RI [22] emerged. Semantic vectors[1] is one of the first package [25] implementing a RI algorithm and defining a negation operator based on quantum logic [24]. Some initial investigations about the effectiveness of semantic vectors for retrieval and filtering tasks are reported in [3] and [16], respectively.

[1] `http://code.google.com/p/semanticvectors/`

The use of a semantic representation for TV programs is presented in [5]. The authors propose a hybrid recommender system based on semantic web technologies for addressing the classical problems of both content-based and collaborative approaches. The definition of a semantic similarity between TV shows is an interesting aspect of that work. Structured knowledge represented by means of ontologies was exploited. Conversely, encyclopedic knowledge is used in our research.

Another interesting approach to add semantics to text is proposed in the Wikify! system [15], which has the ability to identify important concepts in a text (keyword extraction), in order to link these concepts to the corresponding Wikipedia pages. The annotations produced by the Wikify! system can be used to automatically enrich documents with references to semantically related information. The Wikify! approach is similar to that implemented by ESA that we used in our work, even though the latter has been effectively used for several tasks, such as text categorization, semantic relatedness and information retrieval. The most recent result is the ESA-*based* retrieval algorithm, called MORAG, which enriches documents and queries with Wikipedia concepts [8]. The authors proved that a feature selection process has a strong impact on the effectiveness of the algorithm. Recently, ESA has been used for automatic music genre classification, in order to represent music samples through a semantic space model [2], while a preliminary work related to the application of ESA in an information filtering scenario is presented in [18].

Recent advances concerning the future of TV are proposed in the NoTube project [13], which aims to demonstrate how semantic web technologies can be used as a tool to connect TV content and the Web through *linked open data*, as part of the wider trend of TV and Web convergence. Semantic representation of digital content is intended to create more intelligent, responsive and personalised applications, in order to filter interesting programmes and advertising. In that project EPG data are linked to semantic entities in the Linked Open Data cloud. As in our work, for a given TV program a set of related Wikipedia concepts are identified. However, the exploited knowledge source is DBpedia, the structured version of Wikipedia.

7 Conclusions and Future Work

In this work we investigated state-of-the-art machine learning methods in the scenario of TV-show retrieval. We compared a statistical method called *logistic regression*, with an incremental and effective technique for dimensionality reduction based on *random indexing*. The motivation behind the choice of these two approaches is that the former is a gold standard in the text categorization area, and the latter is an effective technique for addressing the scalability problem. We also evaluated the impact of a negation operator based on quantum logic that can model in an effective way negative evidence. We investigated the impact of a knowledge-based feature generation process in order to enhance the classical BOW representation and improve the list of interesting items in a personalization scenario. The best learning method in terms of accuracy was LR.

This shows that LR for information retrieval is also effective in situations where text descriptions are very short and where classes may have only few training examples. Furthermore, the Wikipedia-based enrichment process improved the ranking of the retrieved list of TV shows. These results might be efficiently integrated in the platform presented in Section 2, obtaining more accurate personal channels.

In future work, we will investigate the impact of the quantum negation operator to the vector space without any dimensionality reduction and we will try to generalize the results attained by LR, by carrying out an experimental evaluation on videos available in online video repositories such as YouTube. Furthermore, we will adopt an approach where the number of generated features depends on the text length. Finally, the presented model considers only the user preferences expressed in terms of one liked program genre. In the future we will merge the list of interesting items belonging to different program genres, allowing to assign a different weight to each one.

References

1. Ardissono, L., Kobsa, A., Maybury, M.: Personalized Digital Television: Targeting Programs to Individual Viewers. Human-Computer Interaction Series, vol. 6. Kluwer Academic Publishers, Norwell (2004)
2. Aryafar, K., Shokoufandeh, A.: Music genre classification using explicit semantic analysis. In: Proceedings of the 1st International ACM Workshop on Music Information Retrieval with User-centered and Multimodal Strategies, MIRUM 2011, pp. 33–38. ACM, New York (2011)
3. Basile, P., Caputo, A., Semeraro, G.: Semantic vectors: an information retrieval scenario. In: Proceedings of the 1st Italian Information Retrieval (IIR) Workshop, Padua, Italy, January 27-28. CEUR Workshop Proceedings. CEUR-WS.org (2010)
4. Berry, M.W., Drmac, Z., Jessup, E.R.: Matrices, Vector Spaces and Information Retrieval. SIAM Review 41(2), 335–362 (1999)
5. Blanco-Fernández, Y., Arias, J.J.P., Gil-Solla, A., Cabrer, M.R., Nores, M.L., Duque, J.G., Vilas, A.F., Redondo, R.P.D., Muñoz, J.B.: Avatar: Enhancing the Personalized Television by Semantic Inference. International Journal of Pattern Recognition and Artificial Intelligence 21(2), 397–421 (2007)
6. Chorianopoulos, K.: Personalized and mobile digital TV applications. Multimedia Tools and Applications 36(1-2), 1–10 (2008)
7. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. JASIS 41(6), 391–407 (1990)
8. Egozi, O., Markovitch, S., Gabrilovich, E.: Concept-based information retrieval using explicit semantic analysis. ACM Transactions on Information Systems 29(2), 1–34 (2011)
9. Ehrmanntraut, M., Härder, T., Wittig, H., Steinmetz, R.: The personal electronic program guide - towards the pre-selection of individual TV programs. In: CIKM, pp. 243–250. ACM (1996)
10. Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., Lin, C.-J.: LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research 9, 1871–1874 (2008)

11. Gabrilovich, E., Markovitch, S.: Wikipedia-based semantic interpretation for natural language processing. J. Artif. Intell. Res. (JAIR) 34, 443–498 (2009)
12. Iyengar, S.S., Lepper, M.R.: When choice is demotivating: Can one desire too much of a good thing? Journal of Personality and Social Psychology 79(6), 995–1006 (2000)
13. Nixon, L., Aroyo, L., Miller, L.: NoTube: the television experience enhanced by online social and semantic data. In: 1st International Conference on Consumer Electronics (ICCE 2011) (2011)
14. Merialdo, B., Lee, K.T., Luparello, D., Roudaire, J.: Automatic construction of personalized TV news programs. In: Proceedings of the Seventh ACM International Conference on Multimedia (Part 1), MULTIMEDIA 1999, pp. 323–331. ACM, New York (1999)
15. Mihalcea, R., Csomai, A.: Wikify!: linking documents to encyclopedic knowledge. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, CIKM 2007, pp. 233–242. ACM, New York (2007)
16. Musto, C.: Enhanced vector space models for content-based recommender systems. In: Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys 2010, pp. 361–364. ACM, New York (2010)
17. Musto, C., Semeraro, G., Lops, P., de Gemmis, M.: Random Indexing and Negative User Preferences for Enhancing Content-Based Recommender Systems. In: Huemer, C., Setzer, T. (eds.) EC-Web 2011. LNBIP, vol. 85, pp. 270–281. Springer, Heidelberg (2011)
18. Narducci, F., Semeraro, G., Lops, P., de Gemmis, M.: Explicit semantic analysis for enriching content-based user profiles. In: Proceedings of the 2nd Italian Information Retrieval (IIR) Workshop, Milan, Italy, January 27-28. CEUR Workshop Proceedings, vol. 704. CEUR-WS.org (2011)
19. Pronk, V., Korst, J., Barbieri, M., Proidl, A.: Personal television channels: simply zapping through your PVR content. In: Proceedings of the 1st International Workshop on Recommendation-based Industrial Applications, RecSys 2009 (2009)
20. Rice, J.A.: Mathematical Statistics and Data Analysis. Duxbury Press (2006)
21. Rocchio, J.: Relevance Feedback Information Retrieval. In: Salton, G. (ed.) The SMART Retrieval System - Experiments in Automated Document Processing, pp. 313–323. Prentice-Hall, Englewood Cliffs (1971)
22. Sahlgren, M.: An introduction to random indexing. In: Methods and Applications of Semantic Indexing Workshop, TKE 2005 (2005)
23. Smyth, B., Cotter, P.: Personalized electronic program guides for digital TV. AI Magazine 22(2), 89–98 (2001)
24. van Rijsbergen, C.J.: The Geometry of Information Retrieval. Cambridge University Press, Cambridge (2004)
25. Widdows, D.: Orthogonal negation in vector spaces for modelling word-meanings and document retrieval. In: ACL, pp. 136–143 (2003)
26. Zhang, T., Oles, F.J.: Text categorization based on regularized linear classification methods. Information Retrieval 4, 5–31 (2000)