Inf Syst Front (2018) 20:1157–1171
DOI 10.1007/s10796-017-9780-0

TV-Program Retrieval and Classification: A Comparison of Approaches based on Machine Learning

Fedelucio Narducci[1] · Cataldo Musto[1] · Marco de Gemmis[1] · Pasquale Lops[1] · Giovanni Semeraro[1]

Published online: 29 August 2017

Abstract Electronic Program Guides (EPGs) are systems that allow users of media applications, such as web TVs, to navigate scheduling information about current and upcoming programming. Personalized EPGs help users to overcome information overload in this 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 paper we introduce the concept of personal channel, on which Personalized EPGs are grounded, that provides users with potentially interesting programs and videos, by exploiting program genres (documentary, sports, ...) and short textual descriptions of programs to find and categorize them. We investigate the problem of adopting appropriate algorithms for TV-program classification and retrieval, in the context of building personal channels, which is harder than a classical retrieval or classification task because of the short text available. The approach proposed to overcome this problem is the adoption of a new feature generation technique that enriches the textual program descriptions with additional features extracted from Wikipedia. Results of the experiments show that our approach actually improves the retrieval performance, while a limited positive effect is observed on classification accuracy.

✉ Fedelucio Narducci
fedelucio.narducci@uniba.it

Cataldo Musto
cataldo.musto@uniba.it

Marco de Gemmis
marco.degemmis@uniba.it

Pasquale Lops
pasquale.lops@uniba.it

Giovanni Semeraro
giovanni.semeraro@uniba.it

[1] Department of Computer Science, University of Bari Aldo Moro, Bari, Italy

Keywords Recommender systems · Electronic program guides · Content-based filtering

1 Background and Contribution

1.1 Electronic Program Guides Personalization

The advent of digital television and the availability of a new generation of TV services has led to an unprecedented level of program choice, which constitutes a new instance of the information overload problem. A partial solution is represented by Electronic Program Guides (EPGs), which provide users of television and other media applications with continuously updated menus displaying broadcast programming or scheduling information for current and upcoming programming. The solution is not effective when the EPG is simply an electronic equivalent of the printed guide, with no form of personalization able to provide users with individual suggestions matching their needs and preferences. Consequently, a fully personalized EPG is supposed to analyze user's behavior (her watching history, in this specific scenario) in order to discover her interests, which are included in a personal profile and exploited to recommend the right programs at the right times. This type of EPGs removes the traditional channel boundaries, by providing users with personalized channels, which include only programs fitting their profiles. Typically, recommendation technologies are exploited to this purpose since they implement information filtering techniques able to suggest items of interest to users based on their implicit or explicit preferences.

An example of personalized EPG is Watchmi[1] by APRICO Solutions,[2] a software company part of Philips Electronics, whose mission is to develop video recommender and targeting technology, primarily for the broadcast and Internet industries. Watchmi is available in three forms: online,[3] as plug-in for Microsoft Windows Media Center, and embedded in the Eviado One HD-Receiver for satellite and cable TV. The EPG seamlessly integrates TV and Internet content, learning from the user interaction and recommending shows and videos that match the user's preferences. A screenshot of the Watchmi plug-in for the Microsoft Windows Media Center is shown in Fig. 1.

A user can create a personal channel by selecting any TV program or Internet video asset as *seed*. Based on the seed attributes (such as the program type), similar programs and 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 basic architecture of a personal channel is depicted in Fig. 2, which shows two personal channels ("My Action Movies" and "My Science Documentaries"). They are built by means of filters that retrieve programs and videos based on the characteristics of the seed and recommenders, that "personalize" the channel constantly by re-ranking retrieved items according to feedback provided by the user during the interaction with the channel itself. Usually, the feedback has the form of explicit ratings provided through a discrete scale.

Personal channels allow delivering of user-targeted information, as long as the viewing preferences of individual users are acquired both at a coarse-grained level (e.g., program types, such as SPORT) and at a more fine-grained level (specific interests within program types, such as FOOTBALL MAGAZINES).

This view of a personalized channel can be placed in the more general personalization scenario discussed in (Amatriain and Basilico 2015), where a generic architecture that handles large volumes of existing data and is adaptive to user behavior, is depicted. The main components of the architecture contains one or more machine learning algorithms that allow to build models used during the actual computation of recommendation results. In the following section we analyze the main issues related to the filtering component, the core of the architecture, since it has the responsibility to select items that feed the recommender.

[1] http://www.watchmi.tv/en

[2] http://www.aprico.tv

[3] http://www.watchmi.tv/en/watchmi-search

1.2 Classification and Retrieval of TV Programs

In the scenario depicted in Fig. 2, building personal channels involves two tasks:

1. *retrieval* of TV programs based on the type of the program selected as seed;
2. *recommendation*, i.e. re-ranking of retrieved programs in order to produce the recommendation list for the user. The ranking is usually performed by computing similarity between program descriptions and specific preferences within the user profile.

In this paper, we focus on problems related to the retrieval step. The first one is TV-program classification, that consists into *automatically* assigning every available TV show with one or more program types. In fact, programs can actually come from different sources (e.g., digital TV, IPTV, YouTube, etc.), and the very large number of these multimedia objects makes infeasible the manual assignment of program types. The second problem is the definition of a retrieval model for searching TV programs related to the seed program type, and for ranking the corresponding result set according to user preferences. For solving those problems, we assume that the only information available associated with TV programs is a *short* textual description that describes their content. Therefore, the third problem is the choice of an appropriate representation model suitable to deal with short program descriptions.

Given a set $P = \{t_1, \ldots, t_m\}$ of program types, the two problems can be formally defined as follows:

- **TV program Classification**: given a program p and the corresponding textual description d, select the program type $t \in P$ in which p can be categorized, according to the description;
- **TV program Retrieval**: given a set S of program descriptions and a program type $t \in P$, return a ranked list of k program descriptions from S that best match t.

1.3 Contributions

In this paper we address the following research questions, issued by the above mentioned problems:

1. Which representation model should be adopted for program descriptions that feed classification and retrieval algorithms?
2. Which technique should be used to classify TV programs, provided that only a short textual description is available?
3. Which technique should be used to retrieve TV programs belonging to a given program type?

In order to propose a solution to the first problem, in the following section we describe a novel program representation

Fig. 1 Watchmi plug-in

technique, based on feature generation, that exploits the knowledge stored in Wikipedia to enrich the program descriptions with new information useful for both classification and retrieval tasks. The proposed technique extends the classical Bag-of-Words model, based on keywords, with Wikipedia concepts (i.e. articles). As for the other problems, our contribution consists of a thorough evaluation of some classification and retrieval algorithms (described in Section 4) in order to identify those that perform better in the context of EPG personalization. A comparison with related research is proposed in Section 2, while details of the experiments are presented in Section 5. Finally, conclusions are drawn in the last section, together with directions of future research.

2 Related Work

The problems addressed in this paper relate to relevant work in different research areas: Personalized television (PTV), video classification and retrieval, semantic representation

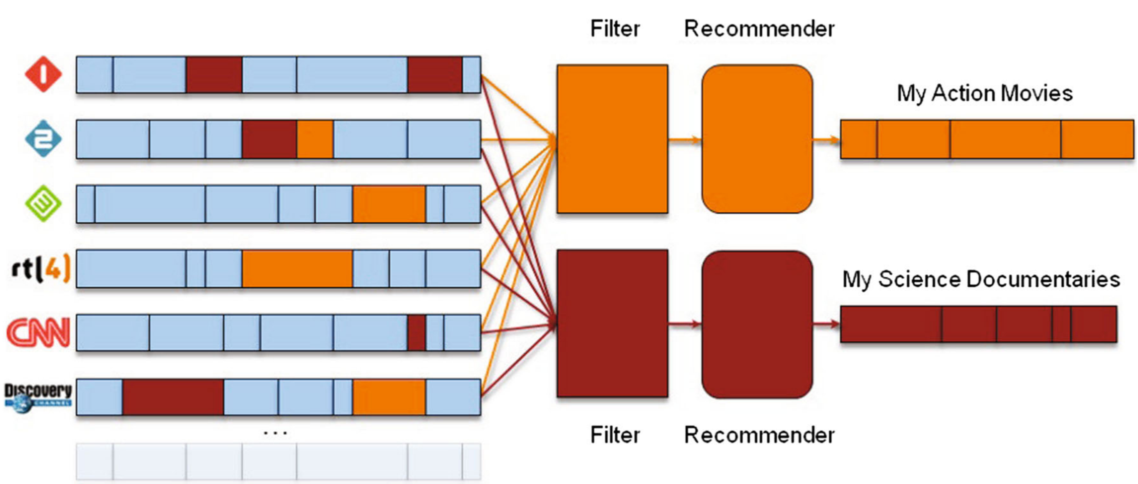


Fig. 2 The concept of personal channel. On the *left* there are the channel sources from which the tv shows used for composing the personal channel are extracted. The filters work by selecting only the tv shows which match with the user seeds. The recommender performs the ranking of the filtered tv shows according to the user feedback

of multimedia content. The first attempts in applying recommendation techniques in the TV domain date back to 20 years ago (Ehrmantraut et al. 1996). The problems addressed in those work mainly regard the user profile to use, and the recommendation technique to apply. Those problems are still considered relevant in the current literature. One of the early personalized EPGs was PTVPlus (Smyth and Cotter 2001), which adopted a hybrid collaborative and case-based approach to suggest programs in the Físchlár (Smeaton et al. 2001) video library system. In this work, the authors mainly addressed the sparsity problem of collaborative systems. Actually, limited availability of user preferences is a critical issue of recommender systems for TV. In (Xu et al. 2002), authors evaluated both explicit and implicit techniques for acquiring user interests, as well as their effectiveness with collaborative and content-based approaches.

A possible solution to overcome this lack of preference information is to use hybrid recommendation techniques that combine collaborative and content-based paradigms, especially for solving the new item problem (Martinez et al. 2009; Cremonesi et al. 2011). In fact, program descriptions can be exploited to find similar programs based on some attributes, such as program types, even if a low number of ratings is available. The YouTube video recommendation system (Davidson et al. 2010) considers video metadata, such as title or description, as well as user activity data, in order to find videos that a user is likely to watch after having watched a given video v. Similar videos are grouped by association rules that allow to compute relatedness among videos, while personalized recommendations are generated by combining co-visitation counts with specific information about user preferences, such as videos that were rated or added to playlists. A more recent approach consists into extracting information about preferred TV programs published by users on their social networks such as Facebook (Shapira et al. 2013) and then use this data together with explicit ratings to improve recommendation accuracy.

In our approach, program information is exploited for selecting the relevant set of programs on which personal channels are built.

The second aspect addressed by our work regards the classification and retrieval of multimedia content. For these tasks, content-based approaches are mainly adopted in literature (Hu et al. 2011). In (Yuan 2003), a generic framework for visual content-based video indexing and retrieval is proposed. This general architecture provides a classification component that comes before the retrieval module. The classification can be related to video genres, video events and objects in the video. More interesting for our goals is the video genre classification. Approaches exploited for this task can be classified into statistic-based, rule or knowledge-based, and machine-learning based. The first class of approaches performs statistical analysis on colors, cuts, camera motion, object motion, and/or other dynamic features. These properties are then exploited for generating more abstract film style attributes. The detected film style attributes are classified into film genres (Fischer et al. 1995). In rule or knowledge-based approaches, heuristic rules from domain knowledge are applied to low-level features in order to classify videos (Hu et al. 2011).

Low-level features are exploited in (Deldjoo et al. 2016), where the authors propose a novel content-based technique that filters items according to stylistic features (lighting, color, and motion) extracted automatically from video files, either full-length videos or trailers, without relying on any high-level features, such as genre, cast, or reviews. The main outcome of this work is that recommendation accuracy is higher when using the considered low-level visual features than when high-level data are employed. This conclusion shouldn't reduce the importance of explicit semantic features in content-based recommender systems. Conversely, these findings provide a powerful argument for exploring both types of features in classification and filtering tasks. Finally, machine learning-based approaches train a classifier or a set of classifiers by using samples described by low-level features. Different classifiers are generally exploited: Bayesian networks (Mittal and Cheong 2004), SVMs (Qi et al. 2006), decision trees (Yuan et al. 2006). Our approach falls in the machine learning-based category. However, the main difference with other solutions presented in literature is that in those work content-based features are extracted directly from video content. This is mainly due to the lack of textual descriptions of multimedia objects, that we try to overcome by adding features generated by external knowledge sources.

The semantic representation of multimedia content is another topic broadly investigated in literature. This task is strictly related to video annotation (Tang et al. 2009; Xu et al. 2008; Snoek et al. 2006), which identifies semantic concepts (such as person, car, people walking) in video shots (Hu et al. 2011), in order to perform video categorization (Ballan et al. 2011). Semantic annotation is mainly based on domain ontologies, often associated to logical-based methods (Shet et al. 2005; Dousson and Le Maigat 2007; Paschke and Bichler 2008; Artikis et al. 2010), but other types of knowledge sources are used as well. For instance, in (Memar et al. 2013) the authors describe a model for semantic-based video retrieval which exploits both WordNet and Columbia374 (Yanagawa et al. 2007) (one of the largest concept detectors for semantic video annotation) to extract semantic concepts from the query. While WordNet provides linguistic knowledge, Columbia374 provides *visual* concepts such as person, waterfront, or explosion, selected from the LSCOM ontology (Kennedy and Hauptmann 2006).

More recent approaches exploit knowledge available on the web, instead of relying on specific ontologies. In (Ko et al. 2014), Linked Data are the source to find the appropriate semantics of the contents extracted from the viewing history of users of a real-world mobile IPTV service. Concepts retrieved for the contents are then grouped together into semantic clusters based on their similarity and relevance, and potentially interesting contents are recommended to general users based on the content-consumption trends monitored from leading user groups who most proactively and frequently consume contents. In (Pappas and Popescu-Belis 2015), similarly to our approach, ESA is adopted as indexing method for titles and descriptions of TED lectures. A pure content-based recommendation method shows that a representation of items based on external knowledge is significantly more useful than the domain knowledge captured intrinsically by the other semantic methods.

Conversely, in our approach ESA is not used as a simple indexing method, but it allows the generation of *new features*. Therefore, original program descriptions are extended with keywords belonging to the semantic interpretation vectors of the most representative Wikipedia articles associated they are associated with. This paper extends the work presented in (Musto et al. 2012). In that work the authors only addressed the tv-show retreival scenario.

3 Techniques for Program Representation

3.1 Bag-of-Words Model

The Bag-of-Words model is the simplest way to represent textual data, such as the description of a TV program. The main idea behind this model is to describe each item by simply listing the words that appear in the text. Given an item i, described by features $\langle f_1, ..., f_n \rangle$, the corresponding BOW representation of item i is:

$$bow_i = \{(f_1, w_1), (f_2, w_2), \ldots, (f_n, w_n)\}$$

where w_k is the weight for the word (feature) f_k. The weight can be computed by different weighting schemes, ranging from the simple boolean scheme, based on simple counting of the occurrences (even normalized) of features in the documents, to the more complex TF-IDF (Baeza-Yates and Ribeiro-Neto 1999).

The classical BOW representation can be also improved by exploiting Natural Language Processing (NLP) techniques, such as stopwords removal, part-of-speech tagging, and stemming (as in (Porter 1980)).

The application of NLP techniques does not guarantee proper content representation, regardless of the type of processed documents. For example, a large corpus of documents usually requires feature selection, with the aim of filtering out irrelevant features, such as very common words. On the other hand, when short documents must be processed, as in the scenario of EPG personalization, feature generation techniques could be adopted to extend and enrich the representation with additional features related to the original content. However, feature generation is not simple to perform, since information changes over time. For example, if we enrich a documentary about the White House with the name of the current US President, that information will be outdated in the future. As a consequence, a recent trend is to exploit open knowledge sources, such as Wikipedia, in which information is constantly updated.

3.2 ESA-based Bag-Of-Words

The ESA-based Bag-of-Words (E-BOW) model is based on a feature generation process that exploits Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch 2006) to associate a program description with a set of related *concepts* (articles) extracted from Wikipedia. These new concepts are included in the original BOW corresponding to the program. The idea is to exploit exogenous knowledge coming from an external concept repository, instead of relying only on the endogenous knowledge obtained from the item descriptions.

The main insight behind ESA is that a possible way to describe the meaning of a term (e.g. *computer*) is to provide a list of concepts it is related to (e.g. *Alan_Turing*, *Artificial_Intelligence*, *mouse*).

When Wikipedia is adopted as a concept repository, a term can be represented by means of its relationships with Wikipedia articles. As a consequence, a fragment of text, such as a program description, can be represented by the set of Wikipedia articles most related to the terms it consists of.

Formally, Wikipedia is seen a large corpus of documents $D = \{d_1, d_2, ..., d_m\}$, which defines a set of concepts $C = \{c_1, c_2, ..., c_n\}$, each one identified by the title of the corresponding article.

For example, the Wikipedia article at: http://en.wikipedia.org/wiki/Artificial_intelligence defines the concept "Artificial Intelligence", identified by the title of the page. It is worth to note that not all Wikipedia pages have a corresponding concept, since there are some pages (e.g. disambiguation pages, pages related to dates, etc.) which are filtered out from this process.

Relationships between terms and concepts in the Wikipedia corpus D are represented by a matrix T, called ESA-*matrix*, in which each column corresponds to a concept, while each row corresponds to a term (word) that occurs in D. The cell $T[i, j]$ holds the TF-IDF value of term t_i in document d_j, which represents the strength of the association between t_i and concept c_j provided by d_j (see Fig. 3).

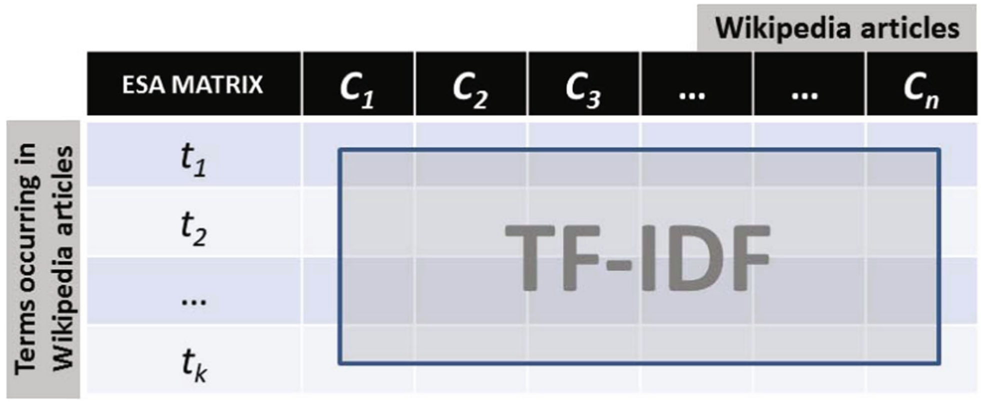


Fig. 3 The ESA-*matrix*

Given a term t_i, the corresponding row in T defines the *semantic interpretation vector* for that term: $\vec{s_i} =< w_{i1}, w_{i2}, , ..., w_{in} >$. In other words, the semantic interpretation vector represents a term in the space of concepts defined by Wikipedia. Following this idea, it is possible to represent any text fragment d in this space as well, by computing the centroid (average vector) $\vec{s} =< w_1, w_2, , ..., w_n >$ of the semantic interpretation vectors associated with words in d. Coordinates are computed as follows:

$$w_j = \frac{\sum_{t_i \in d} \#_d(t_i) \cdot T[i, j]}{length(d)}, \quad (1)$$

where t_i are the keywords in d, $\#_d(t_i)$ is the number of occurrences of t_i in d, $T[i, j]$ is the value stored for t_i and concept c_j in the ESA-*matrix*, and $length(d)$ is the number of keywords in d.

The feature generation process performs basic NLP operations (tokenization, stopword removal, stemmig) on a program description to obtain the corresponding BOW. Then, the semantic interpretation vectors associated with keywords in the BOW are processed as described in Eq. 1, so that the program description is represented in the space of Wikipedia concepts. The most representative concepts, i.e. those with the highest scores, are considered for feature generation. Figure 4 shows an example (in German, because the dataset for the experiments was provided by Philips Research Eindhoven and Axel Springer, see Section 5) of the process for the TV program titled *Rad an Rad - Die besten Duelle der MotoGP (Wheel to wheel - The best duels in the MotoGP)*, related to the *Sport* category. New concepts are associated to the BOW by means of the ESA algorithm: some of them refer to MotoGP riders (Valentino Rossi, Max Biaggi, Loris Capirossi, Shin'ya Nakano), others to MotoGP competitions (großer preis von italien - Italian Grand Prix, großer preis von malaysia - Malaysia Grand Prix). The E-BOW built by the feature generation process consists of the BOW associated with the TV program, augmented by the new concepts (keywords in the titles) identified by the centroid vector of the program description, which hopefully will help in the classification task.

3.2.1 Enhanced Vector Space Model

The Vector Space Model (VSM) is a well-known technique for representing textual documents in a vector space, mainly used in information retrieval and information filtering (Baeza-Yates and Ribeiro-Neto 1999). Given a corpus of documents, each document d is represented as a point in a *n-dimensional* vector space, where n is the number of the distinct terms (features) that occur in the whole corpus:

$$\vec{d} =< w_1, w_2, , ..., w_n >$$

where w_i is the weight of term t_i in document d. Obviously, $w_i > 0$ only for those t_i occurring in the BOW associated with d. Classical similarity measures, such as *cosine similarity*, are adopted to compute closeness between documents.

We proposed an Enhanced Vector Space Model (eVSM), an evolution of VSM in which documents are represented in a *semantic* vector space based on Discriminative Models (DMs) (Musto 2010; de Gemmis et al. 2015).

DMs rely on a simple insight: as humans infer the meaning of a word by understanding the contexts in which that word is typically used, discriminative algorithms extract information about the meaning of a word by analyzing its usage in large corpora of textual documents. This means that it is possible to infer the meaning of a term (e.g., *leash*) by analyzing the other terms it co-occurs with (*dog, animal,* etc.) (Rubenstein and Goodenough 1965). In the same way, the correlation between different terms (e.g., *leash* and *muzzle*) can be inferred by analyzing the similarity between the contexts in which they are used. These approaches rely on the *distributional hypothesis* (Harris 1968), according to which *"Words that occur in the same contexts tend to have similar meanings"*. This means that words are semantically similar to the extent that they share contexts.

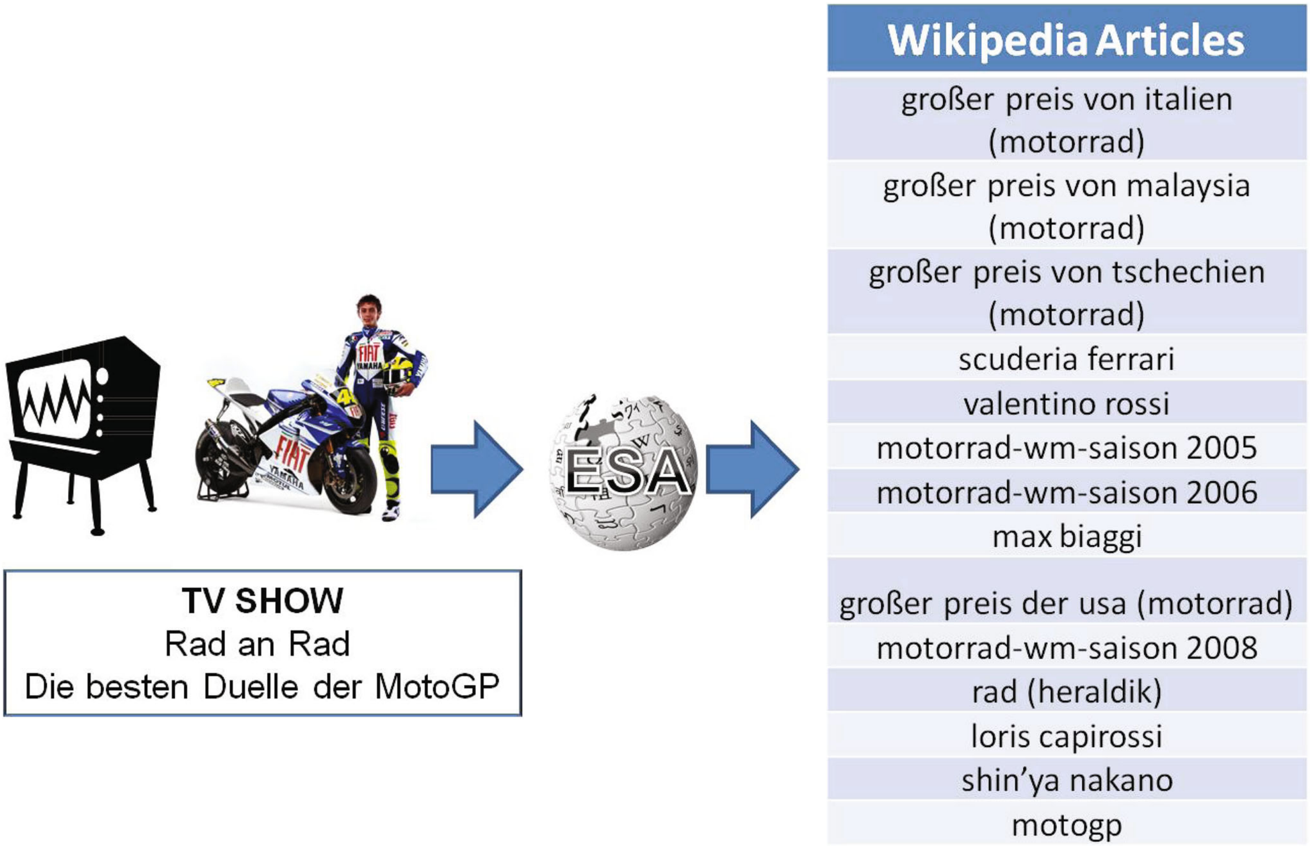


Fig. 4 An example of enrichment by ESA. On the *left* there is the tv show enriched by ESA. On the *right* there are the most relevant concepts from Wikipedia used for enriching the BOW of the tv-show

Differently from the term-concept matrix used by ESA, DMs represent information about terms usage in a *term-context* matrix (Fig. 5), instead of a term-document matrix adopted in the classic VSM. The advantage is that the *context* is a very flexible concept which can be adapted to the specific granularity level of the representation required by the application: for example, given a word, its context could be either a single word it co-occurs with, or a sliding window of terms that surrounds it, or a sentence, or even the whole document. In (Turney and Pantel 2010), it is presented an interesting survey about the three broad classes of VSM to represent semantics, related to the different types of matrix adopted: 1) term-document matrix – usually used to measure similarity of documents, 2) word-context matrix – usually used to measure similarity of terms, and 3) pair-pattern matrix – usually used to measure similarity

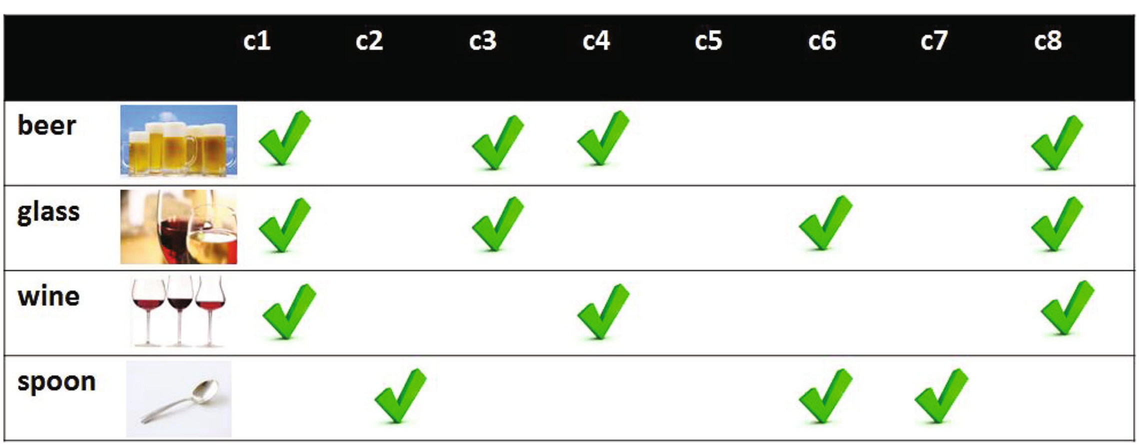


Fig. 5 A term-context matrix. The analysis of the usage patterns of the terms allows to state that *beer* and *wine* or *beer* and *glass* are similar, since they are often used together

of relations (the textual patterns in which the pair X,Y co-occurs, e.g. *X cuts Y* or *X works with Y*).

The classical VSM is the simplest DM proposed in literature, in which co-occurrences are computed by considering the whole document as context. This approach uses *syntagmatic* relations between words to assess their semantic similarity. Indeed, words with a similar meaning will tend to occur in the same document, because they are appropriate to define the particular topic of that document. Instead, the approach based on the co-occurrences computed in a context different from the document uses *paradigmatic* relations, because in a small context window we do not expect that similar words (e.g., synonyms) can co-occur, but we could expect that their surrounding words will be more or less the same.

DMs are referred to as *geometrical models* as well, since each term represented by a row of the term-context matrix can be modeled as a vector. In order to compute relatedness between terms, it is possible to exploit distributional measures that rely on the distributional hypothesis, such as spatial measures (e.g., cosine similarity, Manhattan and Euclidean distances), mutual information-based measures (e.g., Lin), or relative entropy-based measures (e.g., Kullback-Leibler divergence) (Mohammad and Hirst 2012).

On one hand, this representation has the advantage of building a language model, typically referred to as `WordSpace` (Lowe 2001), able to learn similarities and connections in a totally unsupervised way, but on the other hand the dimensionality of vectors when adopting finer-grained representations of contexts is a clear issue (*curse of dimensionality)*. For example, the adoption of sentences as granularity level for contexts causes an explosion of the number of dimensions of the vector space: by assuming 10 to 20 sentences per document on average, the dimension of the vector space would be 10-20 times the one using a classical term-document matrix. For this reason, feature selection or *dimensionality reduction* techniques such as Latent Semantic Indexing (Deerwester et al. 1990) are adopted to transform a high-dimensional space into a lower-dimensionality one.

4 TV-Program Classification and Retrieval

4.1 Classification of TV-Programs

Text categorization or text classification is the activity of labelling natural language texts with thematic categories from a predefined set (Sebastiani 2002). In this section we propose the text categorization algorithms which have been adapted for the task of program classification described in Section 1.2.

4.1.1 Rocchio Method

We propose to adopt eVSM for TV-program classification based on the Rocchio method (Rocchio 1971) for text categorization. It builds an explicit profile or prototypical document of the category c_i (i.e. a program type), which is a weighted list of the terms whose presence or absence is most useful for discriminating c_i. It is worth to note that the categories are completely independent from the context defined in Section 3.2.1. The context allows to represent the word meanings independently from the categories defined in the classification task.

In order to build the prototype vector for the program type t, the Rocchio algorithm needs a set of program descriptions TR_t already associated with p (pre-labeled training examples). The prototype vector p_t is computed as the sum of the vectors which represent documents belonging to TR_t in the eVSM:

$$\mathbf{p}_t = \sum_{d \in TR_t} \mathbf{d}.$$

Given a set of pre-defined program types, the prototype vector is build for each one of them. Then, a TV-program s can be easily classified, by computing the cosine similarity between the prototype vectors of program types and the vector associated with s. The program s is represented as described in the previous paragraph. The category assigned to s is the one having the highest similarity score. In this work the Rocchio Method is only used to build the prototype vector of the program type.

4.1.2 Logistic Regression

Logistic Regression (LR) is a discriminative probabilistic classification model that approximates a real-valued (instead than binary, as in the typical case of classification) function ϕ by means of a function ϕ' that fits the training data. The goal is to learn a model that correctly separate examples belonging to different classes. In particular, the method estimates the probability that the document d_i belongs to the category c_j, and the decision of whether to assign the category can be based on comparing the probability estimate with a threshold or, more generally, by computing which decision gives optimal expected utility.

Even though the gold standard for text classification are Support Vector Machines (Joachims 1998), because of their well-known strenghts (accuracy, robustness, automated tuning of the parameters), LR demonstrated very similar accuracy in the text categorization task. For the categorization of TV-programs we adopt the *one vs the rest* model: a logistic function is learned for each program type. Then, given a TV-program, we compute the probability value for each program type by exploiting the logistic function learned for each class. The TV show is assigned to the program type with the highest probability value.

4.2 TV-program Retrieval

Both eVSM and LR have been adapted to the task of program retrieval as well. As for eVSM, the adaptation follows two steps. First, a semantic vector space based on Distributional Models is built. Next, the prototype vector is built for each program type, as described in the previous section. In the retrieval scenario, given a program type t, the corresponding prototype p_t is used as a query, thus we compute the cosine similarity between p_t and every TV-programs (represented in the semantic vector space). The n TV-programs with the highest similarity score are included in the result set. As regards LR, the probability that a TV show belongs to a specific program type is exploited for the retrieval task as well. Given a program type t, the corresponding learned logistic function is used to compute the probability of belonging to t for all the available TV-programs. The n TV-programs with the highest probability are returned as a result set.

5 Experimental Evaluation

In this section we describe two experiments which have been carried out to answer to the research questions issued in Section 1.3. In particular, the first experiment aims at assessing which are the most effective techniques for the tasks of retrieval and classification, while the goal of the second experiment is to evaluate whether the adoption of the E-BOW representation model for program descriptions improves the predictive accuracy of classification and retrieval algorithms, compared to the standard BOW model.

5.1 Dataset

Both experimental sessions have been carried out on a dataset composed of 133,579 TV shows broadcast by of a set of 47 channels in German language. The dataset was kindly provided by Philips Research Eindhoven and Axel Springer. Table 1 summarizes some statistics about the dataset.

The vocabulary of the dataset, i.e. the number of distinct terms in the program descriptions, is 306,006 (42.11 terms per document on average). Figure 6 shows the distribution of program descriptions on the 17 different types. The dataset is very unbalanced towards very popular program types, such as TV Series (*id: 4*), Movies (*id: 2*) and Documentary (*id: 8*). For other program types, such as Weather (*id: 13*), the number of descriptions available is very low, which can negatively affect the performance of the algorithms.

Table 1 Dataset statistics

TV-programs	Program types	# features (BOW)	avg #cfeatures (TV-program)
133,579	17	306,006	42.11

5.2 Experiment 1

5.2.1 Experimental Protocol and Performance Measures

We adopted *10-fold cross validation* as evaluation protocol. The dataset is partitioned into 10 complementary subsets, then 10 runs of the experiment are performed. In each run, 9 subsets are used for training, the remaining subset for testing the model. For the classification task, training data are used for building the prototype vectors and learning the logistic functions. Then, the classifiers are used to categorize the programs in the test set. The process is then repeated 10 times (the folds), with each of the 10 subsets used exactly once as test data. The effectiveness of the models is evaluated by using the Accuracy *(Ac)* metric, computed as:

$$Ac = \frac{\textsc{CorrectlyClassified}}{\textsc{TotalClassified}}$$

The 10 results from the folds are averaged to produce a single accuracy estimation.

As for the *retrieval* task, training data are used for building the prototype vectors of program types and learning the logistic functions, in the same way as for the classification task. Then, prototype vectors are used as queries, and programs in the test set are ranked according to the value of cosine similarity. As for LR, given a logistic function for a program type t, items in the test set are ranked according to the probability of belonging to t. Performance is measured in terms of Precision@k% (Pr@k%), with $k = \{5, 10, 25, 50, 70, 100\}$, computed as:

$$Pr@k\% = \frac{TP}{\#Ts * k\%}$$

where *TP* are the true positive, and #Ts is the cardinality of the test set for a specific program type. We computed *Pr@k%* separately for each program type and for each of the 10 runs of the evaluation. Finally, we averaged the partial values in order to get the final results, which have been validated by means of the Wilcoxon statistical test. Both classification and retrieval algorithms work on program descriptions represented according to the classic BOW model described in Section 3.1. This allows to define a baseline that can be compared to the proposed representation model based on ESA, described in Section 3.2.

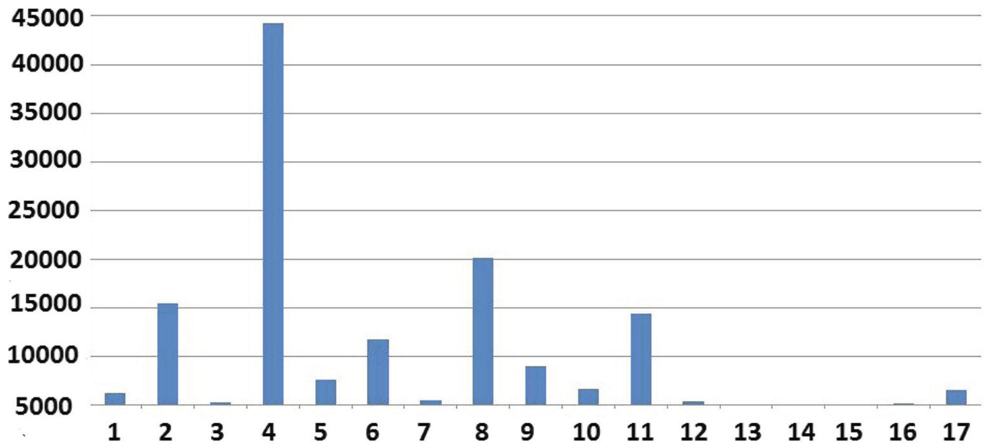


Fig. 6 Distribution of program descriptions among the 17 program types: miscellaneous **(1)**, movies **(2)**, short movies **(3)**, tv series **(4)**, sport **(5)**, show **(6)**, events **(7)**, documentary **(8)**, reportage **(9)**, report **(10)**, magazine **(11)**, news **(12)**, weather **(13)**, videoclip **(14)**, preview **(15)**, advertising **(16)**, music **(17)**

5.2.2 Analysis of the Results

Table 2 shows the results obtained for the classification task by LR and eVSM-Rocchio, splitted per program type (best accuracy in each class is highlighted in bold). The main outcome is that LR outperforms eVSM-Rocchio on 13 out of 15 classes, as well as on average accuracy (0.78 vs. 0.64). If we consider only the classes with a consistent number of training examples (i.e. those having more than 10,000 descriptions; Ids: 2, 4, 6, 8, 11), the results are even more in favor of LR (0.82 vs. 0.47). This could be explained by the fact that LR learns the classification model from both positive examples (those belonging to the target class) and negative examples (programs belonging to other classes), while eVSM-Rocchio builds the prototype vector of a specific class by exploiting only program descriptions in that class. When a lower number of training examples is available (classes having less than 10,000 descriptions; Ids: 1, 3, 5, 7, 9, 10, 12, 14, 16, 17), surprisingly eVSM-Rocchio improves its overall accuracy (+3%), despite the space reduction. A slight decrease of LR performance is observed (-2%), but anyway the algorithm shows robustness with few training data. These results allow us to conclude that LR is the most accurate algorithm, compared also to classic VSM (Musto et al. 2011), even if it is more sensible than eVSM-Rocchio to the lack of training examples.

Table 2 Accuracy of algorithms compared on text classification, reported separately for each program type. Results for *weather* and *preview* are not shown since the number of training examples is too low to build a reliable model

Id	Program Type	eVSM	LR
1	miscellaneous	**0.40**	0.26
2	movies	0.51	**0.83**
3	short movies	**0.89**	0.75
4	tv series	0.72	**0.87**
5	sport	0.87	**0.96**
6	show	0.52	**0.85**
7	events	0.66	**0.86**
8	documentary	0.23	**0.72**
9	reportage	0.33	**0.75**
10	report	0.33	**0.43**
11	magazine	0.36	**0.81**
12	news	0.78	**0.82**
13	weather	—	—
14	videoclip	0.74	**0.83**
15	preview	—	—
16	advertising	0.94	**0.98**
17	music	0.77	**0.84**
—	avg.	0.64	**0.78**

The best values between eVSM and LR are in bold.

Results reported in Table 3 show that LR dominates eVSM-Rocchio in the retrieval task as well. In particular, we observed a decrease of $Pr@k$ for both algorithms, by increasing the number of retrieved items (i.e. by varying k from 5 to 100). Indeed, the worsening of performance is higher for eVSM-Rocchio (from 0.57 to 0.35) than LR (from 0.82 to 0.76), which however maintained a quite satisfying precision, even by considering the whole list of retrieved items (Pr@100%).

The main outcome of this session of experiments is that LR clearly outperformed eVSM-Rocchio in both classification and retrieval tasks.

5.3 Experiment 2

5.3.1 Experimental Protocol and Performance Measures

The aim of the second experiment is to investigate whether the adoption of the proposed E-BOW representation model

Table 3 Results of LR and eVSM-Rocchio on text retrieval (P@k%)

Approach	P@5%	P@10%	P@25%	P@50%	P@75%	P@100%
eVSM	0.57	0.53	0.46	0.41	0.38	0.35
LR	0.82	0.79	0.70	0.71	0.73	0.76

can improve the results of the algorithm that performed better in the previous experiment, namely LR, that worked on the standard BOW model.

The experimental protocol was the same as in the previous experiment. The only difference concerns the representation of programs: given a program, starting from the corresponding BOW, we generated the top-60 most related Wikipedia concepts, given by the feature generation process described in Section 3.2. Then, we performed three different evaluations by adding 20, 40, and all the 60 new features to each BOW. The adopted metrics are the classical Precision, Recall and F-Measure for classification (Sebastiani 2002), $P@k\%$ for retrieval.

5.3.2 Analysis of Results

Tables 4, 5, and 6 show the results for the classification task, divided by program type. Results obtained by using only BOW features are the baseline, while those overcoming the baseline are reported in bold. In general, we observed that the adoption of the E-BOW model does not increase precision (Table 4). The only significant improvement is achieved on the *videoclip* program type, which is one of the classes with the smallest number of training examples and the shortest average length of textual descriptions.

Slightly better results are obtained for Recall (Table 5): E-BOW improves the performance of LR on a larger number of classes, compared to the baseline. The most significant improvements are achieved on *short movies*, *events*, *music*, and again on *videoclip*.

Table 4 Precision comparison between BOW and E-BOW on classification

Id	Program Type	BOW	E-BOW +20	E-BOW +40	E-BOW +60
1	miscellaneous	0.26	0.23	0.25	0.25
2	movies	0.83	0.81	0.82	0.82
3	short movies	0.75	0.65	0.62	0.62
4	tv series	0.87	**0.88**	0.87	0.87
5	sport	0.96	0.94	0.94	0.94
6	show	0.85	0.84	0.85	0.85
7	events	0.86	0.83	0.81	0.82
8	documentary	0.72	0.72	0.71	0.71
9	reportage	0.75	0.72	0.75	0.75
10	report	0.43	0.35	0.39	0.39
11	magazine	0.81	0.80	0.81	0.81
12	news	0.82	0.68	0.71	0.70
13	weather	-	-	-	-
14	videoclip	0.83	**0.85**	**0.88**	**0.87**
15	preview	-	-	-	-
16	advertising	0.98	0.97	0.98	0.88
17	music	0.84	0.78	0.79	0.79
—	avg.	0.78	0.74	0.75	0.74

The values overcoming the baseline (BOW) are reported in bold.

Table 5 Recall comparison between BOW and E-BOW on text classification

Id	Program Type	BOW	E-BOW +20	E-BOW +40	E-BOW +60
1	miscellaneous	0.06	**0.07**	0.06	0.06
2	movies	0.69	**0.70**	0.69	0.69
3	short movies	0.08	**0.11**	**0.09**	**0.10**
4	tv series	0.96	0.95	0.96	0.96
5	sport	0.93	0.93	0.93	**0.94**
6	show	0.80	**0.81**	0.80	0.79
7	events	0.41	**0.44**	0.40	0.41
8	documentary	0.83	0.81	0.82	0.82
9	reportage	0.58	0.58	0.55	0.55
10	report	0.13	0.12	0.11	0.11
11	magazine	0.81	0.79	0.78	0.77
12	news	0.31	0.31	0.31	0.30
13	weather	-	-	-	-
14	videoclip	0.63	**0.67**	**0.69**	**0.68**
15	preview	-	-	-	-
16	advertising	0.96	0.95	0.95	0.86
17	music	0.70	**0.74**	**0.73**	**0.73**
—	avg.	0.59	0.60	0.59	0.58

The values overcoming the baseline (BOW) are reported in bold.

Table 6 F-Measure comparison between BOW and E-BOW on TV-show classification

Id	Program Type	BOW	E-BOW +20	E-BOW +40	E-BOW +60
1	miscellaneous	0.09	**0.10**	**0.10**	**0.10**
2	movies	0.75	0.75	0.75	0.75
3	short movies	0.14	**0.19**	**0.16**	**0.17**
4	tv series	0.91	0.91	0.91	0.91
5	sport	0.95	0.94	0.94	0.94
6	show	0.83	0.82	0.82	0.82
7	events	0.56	**0.58**	0.53	0.55
8	documentary	0.77	0.76	0.76	0.76
9	reportage	0.65	0.65	0.64	0.63
10	report	0.20	0.17	0.17	0.17
11	magazine	0.81	0.79	0.79	0.79
12	news	0.45	0.43	0.43	0.42
13	weather	-	-	-	-
14	videoclip	0.72	**0.75**	**0.77**	**0.77**
15	preview	-	-	-	-
16	advertising	0.97	0.96	0.97	0.87
17	music	0.77	0.76	0.76	0.76
—	avg.	0.64	0.64	0.63	0.63

The values overcoming the baseline (BOW) are reported in bold.

Table 7 Comparison between BOW and E-BOW in retrieval task (Pr@k%)

Pr@	BOW	E-BOW+20	E-BOW+40	E-BOW+60
5%	0.92	0.92	**0.94**	**0.94**
10%	0.90	**0.91**	**0.93**	**0.94**
25%	0.88	**0.90**	**0.92**	**0.93**
50%	0.86	**0.88**	**0.90**	**0.90**
75%	0.82	**0.84**	**0.86**	**0.87**
100%	0.75	**0.76**	**0.78**	**0.79**

The values overcoming the baseline (BOW) are reported in bold.

F-Measure values (Table 6) show significant improvements only on 3 classes: *short movies*, *events* and *videoclip*, while on average results are virtually unchanged.

The main outcome of the experiment is that in general the classification algorithm does not benefit from the adoption of the feature generation process, but improvements are observed on those classes for which poor information (short descriptions, low number of training examples) is available.

Table 7 shows the results of the retrieval task. For the sake of simplicity, we do not detail *Pr@k%* figures for each program type, but we report only averaged values.

The main outcome is that in general E-BOW outperformed the baseline; in particular, better results are obtained when at least 40 most related Wikipedia concepts are added to the BOW. By varying k from 5 to 100, *Pr@k%* values show a decreasing tendency for each one of the evaluated models. However very good results are achieved even for low values of k. This is a valuable finding, since in the EPG personalization scenario depicted in Section 1.1, it is more likely that the system has to retrieve a small number of TV-programs for building a personal channels, rather than suggesting a large list of potentially interesting programs.

We performed the Mann-Whitney test to assess whether the differences between *Pr@k%* for BOW and E-BOW are statistically significant. Given two sets of observations, obtained by two different approaches, and an ordering of those results, the test decides whether the ranked list is achieved by chance or not. We compared the following lists of results:

– BOW VS E-BOW+20 ESA-features
– BOW VS E-BOW+40 ESA-features
– BOW VS E-BOW+60 ESA-features

The result of the test was that all the differences between BOW and E-BOW+40 or E-BOW+60 are statistically significant, while differences between BOW and E-BOW+20 are significant only for *Pr@k%*, $k > 5$ ($p = 0.05$). We can conclude that the ESA-based feature generation process actually improves the precision of the retrieval system when the BOW is extended with the 40 or 60 most related Wikipedia concepts.

6 Conclusions and Future Work

In this paper we investigated the problem of adopting appropriate algorithms for TV-program classification and retrieval in the context of personal channels. This problem is crucially important in this personalization scenario, where typically attributes of a *seed* program, such as its category, are exploited to find similar programs that are then grouped into a personal channel. Logistic Regression was the best approach in terms of classification accuracy. Despite this method already demonstrated its effectiveness in the classical text categorization task, results achieved in this specific scenario, characterized by short documents and a low number of training examples, were not obvious. Experimental results show that Logistic Regression stands out in the retrieval task as well. We analyzed also the impact that the adoption of a semantic approach to enrich program descriptions has on the performance of classification and retrieval algorithms. In particular, starting from the observation that the main problem in this personalization scenario is the lack of program descriptions, we proposed a representation model that exploits exogenous knowledge coming from Wikipedia to integrate keywords in the standard program descriptions. Our ESA-based Bag-of-Words model is able to enrich a program description with a set of related keywords extracted from Wikipedia. Experimental results show that our feature generation process has a positive effect on Logistic Regression when applied to the retrieval task, while the impact on the classification task is quite limited, even if some improvements are observed just on those classes for which poor descriptions are available. As a future work, we will extend the experiments to evaluate other semantic approaches that augment a plaintext with pertinent hyperlinks to Wikipedia pages, such as TAGME (Ferragina and Scaiella 2010). In this context, different strategies to enrich the BOW could be investigated. One limitation of our approach is that the number of concepts to be included in the BOW is predefined (we tested +20, +40, +60); it could be interesting to develop a method that adapts the number of concepts to the length of the text to be enriched.

Acknowledgments This work fulfils the research objectives of the PAC02L1_00061 project MAIVISTO "Massive Adaptive Internet VIdeo STreaming using the clOud" funded by the Italian Ministry of University and Research (MIUR). The authors are grateful to Mauro Barbieri, Jan H. M. Korst, Verus Pronk, Ramon Clout from Philips Research Eindhoven, who provided expertise that greatly supported our research. The authors wish to thank Philips Research Eindhoven and Axel Springer for providing access to the dataset used in the experiments.

References

Amatriain, X., & Basilico, J. (2015). Recommender systems in industry: A netflix case study. In Ricci, F., Rokach, L., Shapira, B., Kantor, P. B. (Ed.), *Recommender systems handbook* (pp. 385–419). Springer.

Artikis, A., Sergot, M., & Paliouras, G. (2010). A logic programming approach to activity recognition. In *Proceedings of the 2nd ACM international workshop on events in multimedia, EiMM '10* (pp. 3–8). New York: ACM. https://doi.org/10.1145/1877937.1877941.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. Boston: Addison-Wesley Longman Publishing Co., Inc.

Ballan, L., Bertini, M., Del Bimbo, A., Seidenari, L., & Serra, G. (2011). Event detection and recognition for semantic annotation of video. *Multimedia Tools and Applications*, *51*(1), 279–302. https://doi.org/10.1007/s11042-010-0643-7.

Cremonesi, P., Turrin, R., & Airoldi, F. (2011). Hybrid algorithms for recommending new items. In *Proceedings of the 2nd international workshop on information heterogeneity and fusion in recommender systems, Hetrec '11* (pp. 33–40). New York: ACM. https://doi.org/10.1145/2039320.2039325.

Davidson, J., Liebald, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U., Gupta, S., He, Y., Lambert, M., Livingston, B., & Sampath, D. (2010). The YouTube video recommendation system. In *Proceedings of the fourth ACM conference on recommender systems, recsys '10* (pp. 293–296). New York: ACM. https://doi.org/10.1145/1864708.1864770.

Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., & Harshman, R.A. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, *41*, 391–407.

de Gemmis, M., Lops, P., Musto, C., Narducci, F., & Semeraro, G. (2015). Semantics-aware content-based recommender systems. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.) *Recommender Systems Handbook* (pp. 119–159). Springer.

Deldjoo, Y., Elahi, M., Cremonesi, P., Garzotto, F., Piazzolla, P., & Quadrana, M. (2016). Content-based video recommendation system based on stylistic visual features. *Journal of Data Semantics*, *2016*(online version), 1–15.

Dousson, C., & Le Maigat, P. (2007). Chronicle recognition improvement using temporal focusing and hierarchization. In *Proceedings of the 20th international joint conference on artifical intelligence, IJCAI'07* (pp. 324–329). San Francisco: Morgan Kaufmann Publishers Inc. http://dl.acm.org/citation.cfm?id=1625275.1625326.

Ehrmantraut, M., Härder, T., Wittig, H., & Steinmetz, R. (1996). The personal electronic program guide - towards the pre-selection of individual TV programs. In *Proceedings of the fifth international conference on information and knowledge management, CIKM '96* (pp. 243–250). New York: ACM. https://doi.org/10.1145/238355.238505.

Ferragina, P., & Scaiella, U. (2010). TAGME: On-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM conference on information and knowledge management, CIKM 2010* (pp. 1625–1628). Toronto: ACM. https://doi.org/10.1145/1871437.1871689.

Fischer, S., Lienhart, R., & Effelsberg, W. (1995). Automatic recognition of film genres. In *Proceedings of the third ACM international conference on multimedia, MULTIMEDIA '95* (pp. 295–304). New York: ACM. https://doi.org/10.1145/217279.215283.

Gabrilovich, E., & Markovitch, S. (2006). Overcoming the brittleness bottleneck using Wikipedia: enhancing text categorization with encyclopedic knowledge, AAAI'06. In *Proceedings of the 21st national conference on artificial intelligence* (pp. 1301–1306): AAAI press.

Harris, Z.S. (1968). *Mathematical structures of language*. New York: Interscience.

Hu, W., Xie, N., Li, L., Zeng, X., & Maybank, S. (2011). A survey on visual content-based video indexing and retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, *41*(6), 797–819. https://doi.org/10.1109/TSMCC.2011.2109710.

Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. In C. Nédellec, & C. Rouveirol (Eds.) *Proceedings of ECML-98, 10th European conference on machine learning, lecture notes in artificial intelligence*, (Vol. 1398 pp. 137–142). Heidelberg: Springer. joachims98.ps.

Kennedy, L., & Hauptmann, A. (2006). LSCOM Lexicon definitions and annotations version 1.0, DTO challenge workshop on large scale concept ontology for multimedia, Tech. rep., Columbia University.

Ko, H.G., Kim, E., Ko, I.Y., & Chang, D. (2014). Semantically-based recommendation by using semantic clusters of users' viewing history. In *International conference on big data and smart computing (BIGCOMP)* (pp. 83–87). *IEEE*.

Lowe, W. (2001). Towards a theory of semantic space. In *Proceedings of the 23rd annual meeting of the cognitive science society* (pp. 576–581).

Martinez, A., Pazos Arias, J., Vilas, A., Duque, J., & Nores, M. (2009). What's on TV tonight? An efficient and effective personalized recommender system of TV programs. *IEEE Transactions on Consumer Electronics*, *55*(1), 286–294. https://doi.org/10.1109/TCE.2009.4814447.

Memar, S., Affendey, L.S., Mustapha, N., Doraisamy, S.C., & Ektefa, M. (2013). An integrated semantic-based approach in concept based video retrieval. *Multimedia Tools and Applications*, *64*(1), 77–95. https://doi.org/10.1007/s11042-011-0848-4.

Mittal, A., & Cheong, L.F. (2004). Addressing the problems of bayesian network classification of video using high-dimensional features. *IEEE Transactions on Knowledge and Data Engineering*, *16*(2), 230–244. https://doi.org/10.1109/TKDE.2004.1269600.

Mohammad, S., & Hirst, G. (2012). Distributional measures of semantic distance: a survey. CoRR arXiv:1203.1858.

Musto, C. (2010). Enhanced vector space models for content-based recommender systems. In *Proceedings of the fourth ACM conference on Recommender systems, RecSys '10* (pp. 361–364). New York: ACM. https://doi.org/10.1145/1864708.1864791.

Musto, C., Narducci, F., Lops, P., Semeraro, G., de Gemmis, M., Barbieri, M., Korst, J., Pronk, V., & Clout, R. (2012). Enhanced semantic TV-show representation for personalized electronic program guides. In Masthoff, J., Mobasher, B., Desmarais, M.C. & Nkambou, R. (Eds.) *Proceedings of user modeling, adaptation, and personalization: 20th international conference, UMAP 2012, Montreal, Canada, July 16-20, 2012* (pp. 88–199). Berlin: Springer. ISBN 978-3-642-31454-4. https://doi.org/10.1007/978-3-642-31454-4_16.

Musto, C., Semeraro, G., Lops, P., & de Gemmis, M. (2011). Random indexing and negative user preferences for enhancing content-based recommender systems. In *E-commerce and web technologies - 12th international conference, EC-web 2011, Toulouse, France, August 30–September 1, 2011. Proceedings, lecture notes in business information processing* Vol. 85 (pp. 270–281). *Springer*.

Pappas, N., & Popescu-Belis, A. (2015). Combining content with user preferences for non-fiction multimedia recommendation: A study on TED lectures. *Multimedia Tools and Applications*, *74*(4), 1175–1197. https://doi.org/10.1007/s11042-013-1840-y.

Paschke, A., & Bichler, M. (2008). Knowledge representation concepts for automated SLA management. *Decision Support Systems*, *46*(1), 187–205. https://doi.org/10.1016/j.dss.2008.06.008.

Porter, M. (1980). An algorithm for suffix stripping. *Program*, *14*(3), 130–137.

Qi, G.J., Song, Y., Hua, X.S., Zhang, H.J., & Dai, L.R. (2006). Video annotation by active learning and cluster tuning. In *Computer vision and pattern recognition workshop, 2006. CVPRW '06* (pp. 114–114). https://doi.org/10.1109/CVPRW.2006.211.

Rocchio, J. (1971). Relevance feedback in information retrieval. In G. Salton (Ed.) *The SMART retrieval system* (pp. 313–323).

Rubenstein, H., & Goodenough, J.B. (1965). Contextual correlates of synonymy. *Communications of the ACM*, *8*(10), 627–633. https://doi.org/10.1145/365628.365657.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM, Computing Surveys*, *34*(1), 1–47.

Shapira, B., Rokach, L., & Freilikhman, S. (2013). Facebook single and cross domain data for recommendation systems. *User Modeling and User-Adapted Interaction*, *23*(2-3), 211–247. https://doi.org/10.1007/s11257-012-9128-x.

Shet, V., Harwood, D., & Davis, L. (2005). Vidmap: video monitoring of activity with prolog. In *Advanced video and signal based surveillance, 2005. IEEE conference on AVSS 2005* (pp. 224–229). https://doi.org/10.1109/AVSS.2005.1577271.

Smeaton, A.F., Murphy, N., O'Connor, N.E., Marlow, S., Lee, H., McDonald, K., Browne, P., & Ye, J. (2001). The Físchlár digital video system: a digital library of broadcast TV programmes. In *Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries* (pp. 312–313). *ACM*.

Smyth, B., & Cotter, P. (2001). Personalized electronic programme guides. *Artificial Intelligence Magazine*, *22*(2), 89–98.

Snoek, C.G.M., Worring, M., van Gemert, J.C., Geusebroek, J.M., & Smeulders, A.W.M. (2006). The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of the 14th annual ACM international conference on Multimedia, MULTIMEDIA '06* (pp. 421–430). New York: ACM. https://doi.org/10.1145/1180639.1180727.

Tang, J., Hua, X.S., Wang, M., Gu, Z., Qi, G.J., & Wu, X. (2009). Correlative linear neighborhood propagation for video annotation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *39*(2), 409–416. https://doi.org/10.1109/TSMCB.2008.2006045.

Turney, P., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research (JAIR)*, *37*, 141–188.

Xu, J., Zhang, L.J., Lu, H., & Li, Y. (2002). The development and prospect of personalized TV program recommendation systems. In *Proceedings of the fourth IEEE international symposium on multimedia software engineering, MSE '02* (p. 82). Washington: IEEE Computer Society. http://dl.acm.org/citation.cfm?id=824463.824803.

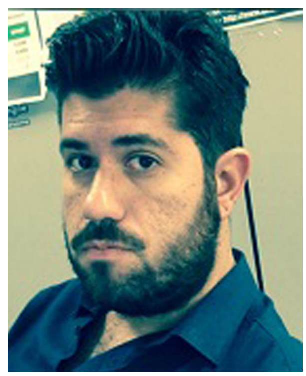
Xu, C., Wang, J., Lu, H., & Zhang, Y. (2008). A novel framework for semantic annotation and personalized retrieval of sports video. *IEEE Transactions on Multimedia*, *10*(3), 421–436. https://doi.org/10.1109/TMM.2008.917346.

Yanagawa, A., Chang, S.F., Kennedy, L., & Hsu, W. (2007). Columbia University's baseline detectors for 374 LSCOM semantic visual concepts. Columbia University ADVENT Technical Report.

Yuan, Y. (2003). *Research on video classification and retrieval*. Xi'an: Ph.D. thesis, School of Electronic and Information Engineering, Xi'an Jiaotong University.

Yuan, X., Lai, W., Mei, T., Hua, X.S., Wu, X.Q., & Li, S. (2006). Automatic video genre categorization using hierarchical svm. In *IEEE international conference on image processing* (pp. 2905–2908). https://doi.org/10.1109/ICIP.2006.313037.

Fedelucio Narducci is post-doc research fellow at University of Bari Aldo Moro, Department of Computer Science. He is also member of the SWAP (Sematic Web Access and Personalization) research group of University of Bari Aldo Moro. His primary research interests lie in the areas of machine learning, recommender systems, user modeling, and personalization. From April 2012 to July 2014 he worked for the SMART (Services & Meta-services for smART eGovernment) project whose goal was to define models, methodologies, languages for planning, production and delivery of services characterized by optimal social value, value of use, and value of exchange. Fedelucio is reviewer and co-reviewer for international conferences and journals on the topics of recommender system, user modeling, and personalization. He is also author of several papers in international journals, conference proceedings, and book chapters.



Cataldo Musto is Research Assistant at the Department of Computer Science, University of Bari. He completed his Ph.D. in 2012 at the University of Bari, under the supervision of Prof. Giovanni Semeraro. His research focuses on the adoption of natural language processing techniques and fine-grained semantics representation for recommender systems and modern adaptive platforms. He has published over 30 papers and served as reviewer in the Program Committee of several conferences in the area.

Marco de Gemmis is Assistant Professor at the Department of Computer Science, University of Bari. He completed his Ph.D. in Computer Science in 2005 at the University of Bari, under the supervision of Prof. Giovanni Semeraro. His current research interests include natural language processing, machine learning techniques for text categorization, information retrieval, and personalized information filtering. Since 2001 he has published around 100 papers.

Pasquale Lops is Assistant Professor at the Department of Computer Science, University of Bari. He completed his Ph.D. in 2005 at the University of Bari, under the supervision of Prof. Giovanni Semeraro. His primary interests lie in the areas of machine learning, recommender systems, digital libraries, user modeling and universal access. He has published over 100 papers and he is involved in several research projects.

Giovanni Semeraro is Associate Professor at the Department of Computer Science, University of Bari. He leads the research group SWAP (Semantic Web Access & Personalization, http://www.di.uniba.it/~swap/) at the Department of Computer Science. His research activity mainly concerns machine learning, semantic web and personalization. He was head of the research units of the University of Bari for the EC 6th Framework Programme Integrated Project VIKEF (2004-07) and DELOS, a Network of Excellence on Digital Libraries (2004-07). He has published over 300 papers.