Enabling Topic-based Discovery of Television Programs

Anugeetha Kunjithapatham, Priyang Rathod, Simon J. Gibbs, Mithun Sheshagiri
Computer Science Lab
Samsung Information Systems America, Inc, San Jose CA 95134
{anugeethak, priyang.rathod, s.gibbs, msheshagiri}@sisa.samsung.com

Abstract

Owing to the often ambiguous and limited metadata available for video content such as television (TV) programs, it is a challenging task for viewers to effectively explore and discover videos of interest from large collections. Our approach to address the above problem is to enable intuitive *topic-based discovery of videos*, whereby users can explore videos related to a topic as well as concepts related to the topic. In this paper, we describe our techniques to achieve the above, which involves semantically enriching video metadata, building a taxonomy of topics and concepts, and finally mapping videos to the taxonomy.

1 Introduction

Videos, in varied forms, such as TV programs and movies, continue to be a popular method to provide and obtain information and entertainment. This is evident in the continual increase in the number of professional and home videos produced every year. For example, historical archives of TV program metadata from Tribune Media Services [TMS] indicate that there has been a 100-fold increase in the number of programs broadcast over the last 50 years..

TV viewers commonly use an on-screen Electronic Program Guide (EPG) to explore broadcast or recorded programs. An EPG includes metadata for the programs such as title, genres, and cast in addition to the date, time and the channel(s) on which the program is broadcast. Using an EPG, a viewer is able to navigate and select content by time, title, genre, actors etc. Some cable operators and TV/Set-top box manufacturers, additionally allow simple keyword-based search for content based on keywords appearing in the various EPG fields. These traditional ways to explore and discover TV content work fine when there are a limited number of programs and channels available for viewers to choose from. However, given that today's cable and satellite viewers typically have hundreds of different channels to choose from at any given time, simple EPG-based navigation is often insufficient.

Some specific limitations of EPG-based navigation include the following: 1) Although program genres, in general, are useful to categorize content, some are ambiguous and misleading. For example, the genre for a program "Earth Diaries: Hindu prayer offerings at Ganges…" is 'Travel'; users might ignore this program based on its genre, even if they are interested in programs about religions. High-level genres such as History and Documentary do not convey anything about the subject of the program. 2) EPG doesn't include information on what kind of topic(s) (for example, 'Person' or 'Location') and related attributes, a particular keyword in the metadata might refer to. 3) Different content producers often use different words to refer to the same entity or related concepts, in the metadata fields (such as title, description) for the programs they produce. For example, suppose one program's description is: "A traditional family meal in Italy", and another program's description is: 'Italian dinner'. If a person is interested in programs about Italian cooking, he would search for the keywords 'Italian' and 'cooking' or 'cuisine' and the above programs would not show up even in the state-of-the-art EPG based program search interfaces.

In our work, we attempt to address the above limitations and enable the following scenario: when a user searches for programs about topic 'A', the interface shows programs related to topic 'A' and programs related to synonyms, related concepts or relevant attributes for topic 'A'. By topic, we mean keywords representing significant entities (for example, 'Barack Obama', 'Iraq War', 'Italian Cuisine' etc.). By attributes, we mean a property of the topic, such as the profession of a 'Person' topic. Essentially, we identify relations/similarities between programs, based on topics appearing in their metadata and concepts related to the topics, and/or synonyms of the keywords representing the topics. In order to achieve this, we need a hierarchy/taxonomy of topics and sub-topics (or, concepts) related to the topics and, we need to map the programs to this topic hierarchy.

Manually creating a taxonomy of topics is not an easy task, as topics relevant to TV programs could vary widely and sub-topics or concepts related to a topic may become more or less relevant over time. Moreover, different users may prefer to categorize content in different ways. On the other hand, the Internet has become an extensive and

valuable source of information and entertainment. More and more people rely on the Internet for information and fort help with everyday activities. Given this, we have come up with techniques to leverage the information provided by the Internet in our effort to enable topic-based discovery of TV programs. We make use of the Internet in primarily two ways: 1) to enrich program metadata, and 2) to build a taxonomy of topics. For metadata enrichment, we use Freebase, the online structured database equivalent of Wikipedia that contains topics of varied categories, as one of the knowledge sources to enrich metadata. We use a variety of web resources including search engines, online directories and dictionaries, to semi-automatically construct the taxonomy of topics and gather related keywords. This topic taxonomy is then used to categorize programs and thereby enable topic-based navigation and discovery.

We describe the above techniques in detail in the following sections. The remainder of the paper is organized as follows: In section 2, we discuss some related work to the different aspects of our approach. In section 3, we describe our approach to: 1) semantically enrich program/video metadata, 2) build a topic taxonomy, and 3) categorize videos by topics. In section 4, we illustrate our prototype system using a few screenshots of the user interface. And, finally we conclude by discussing some benefits of our initial work and plans for future work.

2 Related Work

Several service providers have been looking at ways to facilitate discovery of different types of content (such as videos, photos, bookmarks). Vendors like NetFlix and Hulu provide genre based classification of videos, where some of the genres are hierarchical in nature. Video hosting sites, such as YouTube, prompt users to choose a category, from a set of pre-defined, flat categories when they upload videos. Another approach adopted by YouTube, flickr, del.icio.us and some others, has been to allow users to tag their content while uploading them, and thereby enable keyword-based look up for content. These approaches suffer from the limitations of EPG-based discovery that we discussed in the previous section. Moreover, their systems do not provide a way for users to specify what a tag means or its related topics, nor do they infer these. Some systems such as Twine and Faviki use Wikipedia, a smaller set of locally developed knowledge bases or dictionaries to infer relationships between different pieces of content. In our work, we make use of Freebase, which is a structured and dynamic knowledge source on the web; this enables us to identify many more topics and related concepts corresponding to the keywords appearing in the content metadata, and with better consistency.

Some researchers [Yuan, 2006] have looked at automatically categorizing videos based on an ontology of video genres using hierarchical support vector machines. [Smoliar, 1994] presents tools and techniques for indexing and retrieval of video content. They propose a hierarchical structure for organizing video content and use a frame-based knowledge base to build it. [Wang, 2007] uses multi-modal concepts to represent categories in a latent semantic space and probabilistic latent semantic analysis to classify ad videos into these categories. However, these methods are not able to identify links between videos that are about related topics unless their metadata contains matching terms.

[Yang, 2007] proposes multi-modal approaches for categorizing web videos, including a technique to associate a video with concepts related to words contained in its metadata. They calculate a tf-idf (term frequency–inverse document frequency) score for words appearing in the metadata, and propagate the score to related words, which are identified from WordNet. We have a similar goal to link videos about related concepts, but our approach is independent of the starting set of videos and hence is applicable to different videos.

Many have looked at the problem of organizing text into taxonomies, which is somewhat related to our work on categorizing videos based on topic taxonomies. Some of their approaches range from taxonomy generation for short text segments (keywords) to longer text documents. The idea of using sources on the web to enrich the knowledge required for text, video categorization has been explored by some. [Chuang, 2003] uses terms from search logs to expand domain-specific terminology, which is then used to enrich pre-defined web taxonomies. [Chuang, 2005] presents an approach to create a topic hierarchy for short text segments, by enriching words appearing in the text with the aid of search result snippets. [Pulijala, 2004] and [Dumais, 2000] present methods to classify web documents into a hierarchical structure. Some others [Bloehdom, 2005] have come up with domain-specific (such as, medical) text mining and classification techniques. [Tikk, 2003] presents a method for text categorization using a fuzzy relational thesaurus, a hierarchy that stores an adaptive dictionary at each node in the hierarchy. These local dictionaries are learned incrementally from the document set that needs to be categorized. [Candan, 2008] proposes techniques to organize tag clouds into a hierarchical structure and enable intuitive navigation of blogs.

In summary, the approach commonly adapted to categorize content (videos or documents), is to classify a set of content into a category through various kinds of clustering and classification techniques, which primarily use words appearing in the metadata or text. In our approach, we create a hierarchy of topics and concepts using multiple web sources, in parallel we enrich metadata of videos and finally, map videos to the hierarchy of topics. Thus, the hierarchy of topics used to categorize videos, is built independently of the set of videos that need to be categorized. This enables us to reuse the hierarchy of topics to categorize a different set of videos at any time. Moreover, we calculate a degree of relevance between a video and a topic, which ensures that the most relevant videos are ranked higher and shown first when a user chooses a topic of interest.

3 Our Approach

We address the limitations with EPG based video navigation by enabling topic based discovery of videos. In other words, we enable users to simply choose a topic of interest from a hierarchy of topics, and explore videos related to the topic. This is achieved in three steps:

a) Semantically enrich video metadata: We map keywords appearing in the EPG metadata of TV programs, to topic(s) identified from various sources such as, Freebase and TMS EPG archives. Further, we identify the type and some significant attributes of the topic(s), and add these as additional metadata/secondary topics for the TV program.

b) Build topic taxonomy: We semi-automatically build a taxonomy of topics and sub-topics, and further identify keywords related to each topic and sub-topic in the taxonomy. In this paper, a topic in the taxonomy is sometimes referred to as a 'node' and a sub-topic as a 'child-node'.

c) Categorize videos by topics: Finally, we match topics in the taxonomy to videos, based on the enriched video metadata and keywords related to the topics. The hierarchical relationship between the topics in the taxonomy and how closely the videos are related to the topics and its parent topics are taken into account to identify how relevant a video is to a particular topic.

We describe the above ideas in detail in the following sub-sections.

3.1 Enriching Metadata

By enriching metadata, we mean identifying and adding additional information about the topics discussed in the videos to the video metadata. We use Freebase [Freebase, 2008] and Tribune Media Services's EPG Archives as knowledge sources to identify topics. Freebase is a dynamic, growing online database of topics. It includes several thousands of topics from Wikipedia, in addition to topics entered by general users, Freebase developers & community. TMS EPG Archives includes EPG metadata for TV programs broadcast in the last 50 years. In particular, we made use of the cast and crew information for TV programs in the TMS data and considered them as topics of type 'Person'. For prototyping purposes, we created a local topic database (TopicDB) with topics from Freebase and 'Person' topics from TMS. For data about topics and related attributes available in Freebase, we simply used their data dump.

Our approach to enrich video metadata involves the following steps:

1. Existing metadata for a video content including title, sub-title/episode name and description are retrieved. Closed captions are also retrieved if available. Natural language processing and genre-based keyword extraction techniques [Messer, 2008], are applied on this data to extract significant keywords (single-word) and keyword phrases (multi-word). (We refer to both keywords and keyword phrases as 'keywords' hereafter). By using these techniques, we are able to filter out insignificant keywords (for example, 'the' 'and' etc.). The keywords extracted are then ranked based on two factors: the number of times the keyword appears in the metadata fields and the number of metadata fields (for e.g., title, description) it appears in. To compute f(kw), we identify the number of times a keyword appears in each metadata field, normalize it on a scale of 0-1 and add up the normalized frequencies (corresponding to each metadata field). For keywords extracted from the closed captions, only the top 'N' (say, 10) frequently appearing keywords are considered.

   In short, for keyword 'kw': rank(kw)= f(kw) * n(kw-sources), where f(kw) is normalized frequency of the keyword, and n(kw-sources) is a normalized value (on a scale of 0-1) for the number of metadata-fields the keyword appears in.
2. Then, we match these keywords against topics available in TopicDB. We essentially check if the keywords correspond to any significant topic(s), as identified by the sources used to populate TopicDB. This is done by simply performing a string comparison.
3. If there are multiple topic matches available for the keyword, the first topic is considered as the best match. Once a topic is identified, some useful attributes of the topics are also retrieved from TopicDB and added to the metadata of the video. For example: for 'Person' topics, values for the 'profession' attribute were used.
4. If there are no topic matches available for a keyword 'kw1', a string normalization technique is applied on the keyword to identify any possible spelling errors. This step is done as we have often noticed spelling errors in closed captions and sometimes in the EPG metadata. All topics available in the TopicDB that start with the same letter as 'kw1', and whose length is comparable to the length of 'kw1'are identified as follows: the length of 'kw1' (say, len) is calculated and the starting letter of 'kw1'is identified (say, 'a'); then, 'TopicDB' is queried for all topics starting with 'a' and with length between (len-1) and (len + 1) (say, topicMatchList); and finally, the string distance between 'kw1' and each of the topics identified in the above step are computed. A string distance computing algorithm called 'edit distance' [Edit Distance], is used for this purpose. If the string distance is less than or equal to 2, the topic is considered as a match for the keyword. For example: say, the keyword in hand is 'Bill Cliton' and a topic 'Bill Clinton' exists in the TopicDB, the string distance between these two words is 1, and consequently, the topic 'Bill Clinton' will be considered as a topic match for the keyword. If there are multiple topic matches with the least string distance, we simply choose the first topic match.
5. Keywords with no topic matches, and with a rank below a certain threshold are not associated with the video.

At the end of this process, videos are associated with enriched metadata that essentially is a list of keywords (including topic matches, attributes and other keywords) along with a rank for each of them. This data is stored in a database, say 'VideoDB'.

3.2 Building Topic Taxonomy

We create a taxonomy of topics as the second step in enabling topic-based discovery of videos. Each node in the taxonomy is represented by the tuple: *Node N <name, children, keywords>*, where, 'name' is the topic that the node represents, 'children' is the list of child-nodes that are essentially sub-topics of the topic represented by node N and 'keywords' is a set of keywords (that represent concepts, locations, person names etc.) that are related to the given topic. For example, for a topic 'Italian Cuisine', related keywords are 'pasta', 'pizza', 'spaghetti', etc. Further, we calculate a rank for each keyword in the list, to indicate a degree of relevance of the keyword for the given topic. The taxonomy is created by starting from the root topic node and then adding its children nodes to the taxonomy recursively. This primarily involves: 1) Identifying sub-topics for each node and, 2) Determining representative keywords for each node. At the time of writing this paper, we assumed the availability of the structure of the taxonomy and concentrated on the problem of determining keywords for each node in the taxonomy.

For our prototype, we manually created a taxonomy skeleton with 'History' as the root topic (see Figure 1 in section 4). In order to enrich and extend this taxonomy skeleton, we use several web-based sources from where keywords related to a topic can be extracted. The sources we considered in our prototype are: Wikipedia, Google Directory, Yahoo Directory, Mozilla Directory, WordNet, Dictionary.com, Freebase and the Ask, Google and Yahoo search engines. For the search engines, we considered the related searches and concepts suggested along with search results. For some of these sites we used their web-APIs to obtain keywords, and for others, we developed web scrapers to extract keywords from their web pages. Also, some of the sites provide the keywords directly, while others only provide a textual description, in which case keywords are extracted using natural language processing techniques. Ultimately, for each given topic, we get a list of keywords from these web sources that are directly or indirectly related to the topic. As this list can be fairly long for some topics and possibly contain irrelevant keywords, we further apply some process to trim the list and remove noise. The steps involved in this process are:

1. The use of multiple web-sources for gathering keyword becomes helpful here. We assume that if a keyword occurs in only one source, then the likelihood that it is noise (i.e. not very relevant) is higher. Therefore, all keywords that appear in less than two sources are removed.
2. For each of the remaining keywords, a rank is assigned. The rank of a keyword is based on the number of sources the keyword appears in, the number of times the keyword appeared in each source and the reliability of the source. For example, in our prototype, we consider WordNet and Dictionary.com more reliable than other sources and consequently, rank the keywords that appear in these two sources higher than other keywords. All keywords with a rank lower than a pre-determined threshold are removed. The keywords, thus obtained, along with their respective ranks are stored in the TaxonomyDB.

Further, we personalize the taxonomy in an implicit manner, by using the user's viewing history. For this purpose, a list of keywords is collected from the metadata of the programs watched by the user (say, pKwList). The keywords related to the topic nodes in the taxonomy, which are also present in pKwList are then ranked higher than the rest of the keywords. These ranks are then used to calculate the degree of match between a video and a topic during the Video Categorization process, which is described in the following sub-section. Data about taxonomy nodes including the topic corresponding to the nodes, child nodes and related keywords along with their ranks are stored in a database, 'TaxonomyDB'.

3.3 Categorizing Videos

Keywords associated with each topic node in the taxonomy are compared to the topics/keywords associated with the videos. The number of keywords that are associated (or related) both with a topic node and a video is considered a metric to decide if a video should be mapped to and thereby categorized by the topic node. The specific steps involved in categorizing videos by topics are described below:

1. Topics in the taxonomy are retrieved from TaxonomyDB, and are processed top-down (essentially, parent topics are always processed before the child topics). Following steps are executed for each topic node:
2. List of keywords for topic 'T' (say, 'n' keywords) are retrieved along with their ranks and the following is done:
   a. For each keyword in this list, VideoDB is looked up and all videos that are associated with the same keyword are identified. If no videos are associated with that keyword, the next keyword is processed.
   b. For each video identified in the previous step, a video-keyword rank is calculated based on: the rank of the topic keyword (assigned when the taxonomy is built) and the rank of the video keyword (assigned when video metadata is enriched).
   c. The sum of video-keyword ranks for all videos that match all keywords for topic 'T' is calculated.

$$vtRank_0(V,T) = \sum_{i=1}^{n} vkRank(V, K_i(T))$$

   where, $vtRank_0(V, T)$ is video-topic rank for video 'V' & topic 'T'; $vkRank(V, K_i(T))$ is video-keyword rank for video 'V' and keyword K_i of topic 'T'; 'n' is the total number of keywords related to topic 'T'.
   d. All video matches for topic 'T', along with video-topic ranks are stored in a database, 'VideoTopicRankDB'
3. All parent topics (including parent of parent topics) for topic 'T', if available, are retrieved.
   a. Among videos related to the topic 'T', videos that are also related to it's parent-topics are identified and retrieved along with the video-parenttopic rank, from VideoTopicRankDB.

b. A video-parenttopic rank is weighted based on the distance between the topic 'T' and the parent topic. For example, if parent-topic 'PT_i' is a parent of a parent topic of 'T', distance between them is considered as 2.

$$vtRank(V,T) = \sum_{i=1}^{n} (vtRank(V, PT_i(T) * 1/D(T, PT_i(T)) + vtRank_0(V,T)$$

where, $vtRank_0(V, T)$ is video-topic rank for video 'V' & topic 'T', as calculated in step 2c; $vtRank(V, PT_i(T))$ is video-parenttopic rank for video 'V' and $PT_i(T)$, a parent topic of T; D(T, PTi(T) is the distance between topic 'T' and parent topic PTi(T); 'n' is the total number of parent topics for topic 'T'.

Finally, VideoTopicRankDB contains data on all topic matches for all videos, along with a rank for each video-topic match. This data along with the topic taxonomy available in TaxonomyTB, can be used by a user interface system to enable users to browse through a hierarchy of topics and discover videos related to a particular topic, irrespective of the level of topic in the hierarchy.

4 Prototype

A prototype system ('DeepVideo') based on some of the ideas described in section 2 was implemented using .NET technologies. The system consists of 4 modules: the taxonomy builder, metadata enricher, video categorizer and the user interface module. The UI module was implemented in C# and XAML and the other modules were implemented in C#. SQL server was used for data storage and processing.

We used metadata including title, description, genres, sub-title, and episode name, for about 30K Documentary TV programs to experiment with our prototype. All the programs were of genre 'History'. We specifically considered 'Historical Documentary' videos for our test bed, as our topic-based discovery techniques may fit well with how users explore documentary videos. We created a seed taxonomy with 'History' as the root topic. A snapshot of this taxonomy is provided in Figure 1.

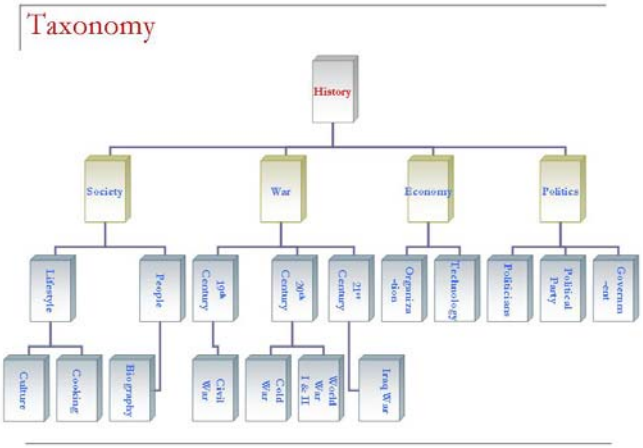


Figure 1: Seed Taxonomy for Prototype

4.1 User Interface Screenshots

In this section, we illustrate our prototype UI and how a user can use DeepVideo by walking through a user scenario: As John switches on his TV, some widgets appear on the left side of the TV screen, as shown in Figure 2. John wishes to explore documentary videos and hence chooses the last widget that corresponds to DeepVideo. A list menu on the left shows some sub-topics under the root topic 'History' and the box on the right shows a tag cloud containing keywords related to the highlighted topic 'Society'.



Figure 2: Top level topics in the taxonomy & related keywords

John is interested in the sub-topics related to 'Politics'. Using the up/down arrow keys on his remote control, he highlights 'Politics' and then presses the right arrow. The UI is refreshed to show a set of sub-topics related to 'Politics' and a tag cloud on the right with keywords related to the highlighted sub-topic 'Political Party', as shown in Figure 3.

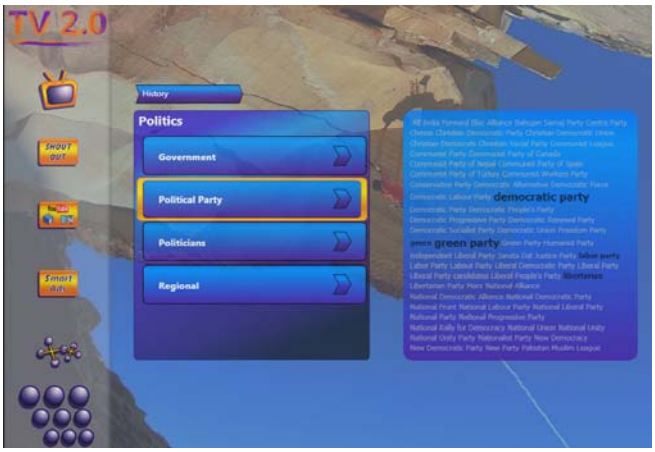


Figure 3: Sub-topics for 'Politics' and tag cloud for 'Political Party'

Then, John chooses the sub-topic 'Politicians' using his remote control. The UI is refreshed to show metadata for videos related to 'Politicians' on the left and a video thumbnail on the right, as shown in Figure 4. The metadata shown includes the video title, episode name and year of broadcast. Due to copyright issues, we have shown a mocked-up thumbnail in the Figure. John browses the list and chooses to watch the video 'History's Mysteries'.



Figure 4: Videos related to topic 'Politicians'

5 Conclusion and Future Work

We described some initial ideas for enabling users to intuitively discover videos based on topics they are interested in. Our techniques can be utilized in several ways, such as to: 1) enhance the EPG-based navigation available today for TV programs and enable better keyword-based search. This would add value to the state-of-the-art STBs/TVs; 2) create a new kind of navigation interface to explore a structured collection of content, as we illustrated using our prototype UI.

While our current approach to build a topic hierarchy is semi-automatic, requiring a user to create/start with a simple seed taxonomy, we plan to explore ideas to automate this step and build taxonomies consisting of multiple hierarchies in the future. Also, we simply map a keyword in the video metadata to the first topic match out of many matches in the current version. We realize from our initial prototype that this could lead to incorrect topic matches and have thought about approaches to implement a topic verification step. One idea is to consider the topic matches for other keywords in the video's metadata and identify if and how many overlapping attributes are found between the topics.

Our observation in general is that the more effort on the part of the program creators or distributors, or the viewing community, to accurately tag/describe a program, the better our topic-based program search results are likely to be. We plan on doing a formal and extensive user evaluation in the near future to further strengthen our approach.

References

[Bloehdom, 2005] Stephan Bloehdorn, Philipp Cimiano, Andreas Hotho. Learning Ontologies to Improve Text Clustering and Classification. *Proceedings of the 29th Annual Conference of the German Classification Society*. Magdeburg, Germany. 2005.

[Candan, 2008] K. Selcuk Candan, Luigi Di Caro and Maria Luisa Sapino. In Proc. of International Workshop on Searching Social media, co-located with the 17th ACM Conference on Information and Knowledge Management. Napa Valley, USA. 2008.

[Chuang, 2003] Shui-Lung Chuang and Lee-Feng Chien. Enriching Web taxonomies through subject categorization of query terms from search engine logs. Decision Support System, 30(1), 2003

[Chuang, 2005] Shui-Lung Chuang, Lee-Feng Chien. Taxonomy generation for text segments: A practical web-based approach. *ACM Trans. Information Systems* 23(4). p363-396. 2005.

[Dumais, 2000] Susan Dumais and Hao Chen. Hierarchical classification of web content. *Proceedings of SIGIR'00*. pp. 256-263. 2000.

[Edit Distance] http://siderite.blogspot.com/2007/04/super-fast-and-accurate-string-distance.html

[Freebase, 2008] Metaweb Technologies, Freebase Data Dumps, http://download.freebase.com/datadumps, Jul'08

[Messer, 2008] Alan Messer, Anugeetha Kunjithapatham, Phuong Nguyen, Priyang Rathod, Mithun Sheshagiri, Doreen Cheng and Simon Gibbs. SeeNSearch: A context directed search facilitator for home entertainment devices. *Pervasive Mobile Computing.* Vol. 4, Issue 6, p 871-888, 2008.

[Pulijala, 2004] Ashwin Pulijala and Susan Gauch, Hierarchical Text Classification. *International Conference on Cybernetics and Information Technologies*. Orlando, FL. 2004.

[Smoliar, 1994] Stephen W. Smoliar, HongJiang Zhang. Content-Based Video Indexing and Retrieval. *IEEE MultiMedia*. vol. 1. no. 2. pp. 62-72. June 1994.

[Tikk, 2003] Domonkos Tikk, Jae Dong Yang, Sun Lee Bang. Hierarchical text categorization using fuzzy relational thesaurus. Kybernetika. 583-600. 2003.

[TMS] Tribune Media Services. http://tms.tribune.com.

[Wang, 2007] Jinqiao Wang , Lingyu Duan , Lei Xu , Hanqing Lu , Jesse S. Jin. TV ad video categorization with probabilistic latent concept learning. In *proceedings of the international workshop on Workshop on multimedia information retrieval*. September 24-29, 2007. Augsburg, Bavaria, Germany.

[Yang, 2007] Linjun Yang, Jiemin Liu, Xiaokang Yang, Xian-Sheng Hua. Multi-modality web video categorization. *Multimedia Information Retrieval*. p265-274. 2007.

[Yuan, 2006] Xun Yuan, Wei Lai, Tao Mei, Xian-Sheng Hua, Xiu-Qing Wu and Shipeng Li. Automatic video genre categorization using hierarchical SVM. In Proceedings of IEEE International Conference on Image Processing, Atlanta, USA, Oct. 2006.