Chapter 5

MEDIA AUGMENTATION AND PERSONALIZATION THROUGH MULTIMEDIA PROCESSING AND INFORMATION EXTRACTION

Nevenka Dimitrova[1], John Zimmerman[2], Angel Janevski[1], Lalitha Agnihotri[1], Norman Haas[3], Dongge Li[4], Ruud Bolle[3], Senem Velipasalar[3], Thomas McGee[1], and Lira Nikolovska[5]

[1] *Philips Research, 345 Scarborough Rd., Briarcliff Manor, NY 10510, USA {Nevenka.Dimitrova,Angel.Janevski,Lalitha.Agnihotri,Tom.McGee}@philips.com)*

[2] *Human-Computer Interaction Institute, Carnegie Mellon, Pittsburgh, PA, USA johnz@cs.cmu.edu*

[3] *IBM T.J. Watson, 30 Saw Mill River Road, Hawthorne, NY 10532, USA {nhaas,bolle}@us.ibm.com*

[4] *Motorola Labs, 1301 East Algonquin Road, Schaumburg, Illinois 60196, dongge.li@motorola.com*

[5] *MIT, Department of Architecture, 265 Massachusetts Avenue N51-340 Cambridge MA 02139, USA lira@mit.edu*

Abstract: This chapter details the value and methods for content augmentation and personalization among different media such as TV and Web. We illustrate how metadata extraction can aid in combining different media to produce a novel content consumption and interaction experience. We present two pilot content augmentation applications. The first, called MyInfo, combines automatically segmented and summarized TV news with information extracted from Web sources. Our news summarization and metadata extraction process employs text summarization, anchor detection and visual key element selection. Enhanced metadata allows matching against the user profile for personalization. Our second pilot application, called InfoSip, performs person identification and scene annotation based on actor presence. Person identification relies on visual, audio, text analysis and talking face detection. The InfoSip application links person identity information with filmographies and biographies extracted from the Web, improving the TV viewing experience by allowing users to easily query their TVs for information about actors in the current scene.

Key words: Content augmentation, personalization, profile, personal news, video indexing, video segmentation, video summarization, information extraction, TV interface, user interface design, interactive TV.

1. INTRODUCTION

For many years, people have enjoyed using their televisions as a primary means for obtaining news, information and entertainment, because of the rich viewing experience it provides. TVs offer viewers a chance to instantly connect with people and places around the world. We call this a *lean-back* approach to content consumption. More recently the Web has emerged as a comparably rich source of content. However, unlike TV, which allows users to select only channels, the Web offers users much more interactive access to expanding volumes of data from PCs and laptops. We call this a *lean-forward* approach to content. We explore the process and value of linking content from these two different, yet related, media experiences. We want to generate a *lean-natural* approach that combines the best of these two media and marries it to users' lifestyles.

At a high level we wanted to explore how cross-media information linking and personalization generates additional value for content. We call this research direction *Content Augmentation*. As an example, imagine a user watches a movie that has characters gambling in Las Vegas. A content augmentation application can extract the location from the movie, then, in anticipation of the user's inquiry, it can peruse the Web for supplemental information such as the prices and availability of rooms in the casino featured in the film, instructions for the game the characters play, information on the design and history of the hotel, etc. In addition, this application can employ a user profile, personalizing the linked content by prioritizing the types of links a user most often explores.

To test this model, we developed a pilot system. We began by focus group-testing several concepts, and, based on the group's reaction, designed and implemented a personal news application (MyInfo) and a movie information retrieval application (InfoSip) that enhances the traditional media experience by combining Web and TV content.

This paper details current TV experience (Section 2.1), related work in content understanding and Web/TV information linking (Section 2.2), our user-centered design process (Section 3.1), pilot applications (Sections 3.2 and 3.3), system overview (Section 4), multimedia annotation and integration methods (Section 5), Web information extraction methods (Section 6), and our personalization model (Section 7). We present our conclusions in Section 8.

2. AUGMENTED USER EXPERIENCE

The current TV experience grows out of a 50-year tradition of broadcasters trying to capture a mass audience. They used both demographic data and input from advertisers to determine which programs to play at the various times of day. More recently, the emergence of niche-based TV channels such as CNN (news), MTV (music), ESPN (sports), and HGTV (home and garden) allows viewers more control over when they view the content they desire. In addition, the arrival of electronic program guides (EPGs) have allowed viewers to browse the program offerings by genres, date, time, channel, title, and, in some cases, search using keywords, a big step forward over traditional paper guides that allow access by time and channel only.

2.1 The Current TV Navigation and Personalization

Current EPGs found in digital satellite settop boxes, cable settop boxes, and personal video recorders from TiVo (www.tivo.com) and ReplayTV (www.digitalnetworksna.com/replaytv/default.asp) offer users advanced methods for finding something to watch or record. These systems generally hold one to two weeks' worth of TV data, including program titles, synopses, genres, actors, producers, directors, times of broadcast, and channels. Viewers can use EPGs to browse listings by time, channel, genre, or program title. In addition, viewers can search for specific titles, actors, directors, etc. Finally, the TiVo system offers a recommender that lists highly rated programs and automatically records these programs when space is available on its hard disk.

Although TiVo is currently the only commercial product with a recommender, much personalization research has been done in this area. Das and Horst developed the TV Advisor, where users enter their explicit preferences in order to produce a list of recommendations (Das et al. 1998). Cotter and Smyth's PTV uses a mixture of case-based reasoning and collaborative filtering to learn users' preferences in order to generate recommendations (Cotter et al. 2000). Ardissono et al. created the Personalized EPG that employs an agent-based system designed for settop box operation (Ardissono et al. 2001). Three user modeling modules collaborate in preparing the final recommendations: Explicit Preferences Expert, Stereotypical Expert, and Dynamic Expert. And Zimmerman et al. developed a recommender that uses a neural network to combine results from both an explicit and an implicit recommender (Zimmerman et al. This

Volume). What all these recommenders have in common is that they only examine program-level metadata. They do not have any detailed understanding of the program, and cannot help users find interesting segments within a TV program.

There has been also research in personalization related to adaptive hypermedia systems (Brusilovsky, 2003). These systems build a model of the goals, preferences and knowledge of each individual user, and use this model throughout the interaction with the user, in order to adapt to the needs of that user.

The Video Scout project we previously developed offers an early view of personalization at a subprogram level (Jasinschi et al. 2001, Zimmerman et al. 2001). Video Scout offers users two methods for personalizing the TV experience. First, Scout can display TV show segments (Figure 5-1). For example, it segments talk shows into host/guest segments, displays musical performances and individual jokes. Second, Scout offers a user interface element called "TV magnets" (Figure 5-2). If users specify financial news topics and celebrity names, then Scout watches TV and stores matching segments, monitoring the contents of talk shows for celebrity clips and searching the contents of financial news programs for financial news stories. Subprogram level access to TV programs improves the TV experience by allowing users more control over the content they watch.

Figure 5-1. Talk show segmented into host and guest segments.

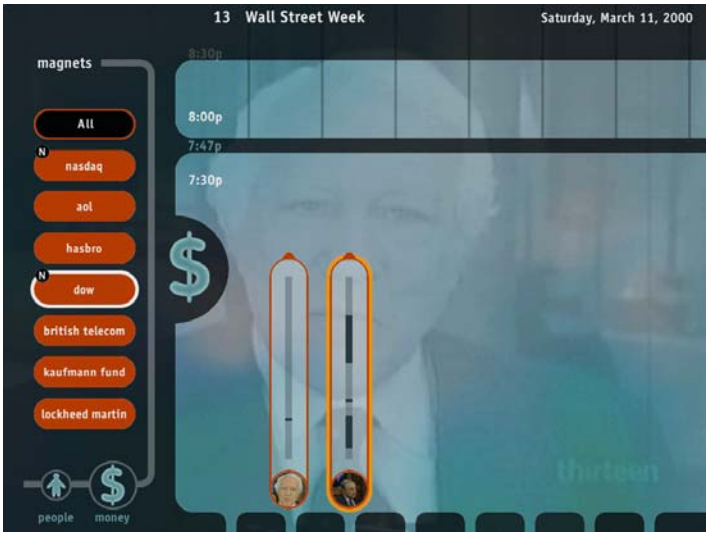

Figure 5-2. Financial news magnet screen with four stored clips from two TV shows.

2.2 Related Work in Content Analysis and Enhanced TV

Recently, there has been increasing interest in hyperlinking video with supplemental information. Examples include Microsoft and CBS's interactive TV (Microsoft 1997), ABC's enhanced TV (ABC 2003), the HyperSoap project at the MIT Media Lab (Dakss), and Jiang and Elmagarmed's work on their Logical Hypervideo Data Model (Jiang et al. 1998).

In 1997 at the National Association of Broadcaster's Expo, we saw Microsoft demonstrate their Enhanced TV concept. This concept allowed users to see Internet data associated with a TV program while watching the program. The Internet content appeared on the side and bottom of the TV screen while the TV show played. Since then Microsoft has been working with broadcasters such as CBS to deliver interactive TV versions of the Grammy Awards, NCAA Basketball, and even TV dramas like CSI (Microsoft 2000). The current implementation works only for users with WebTV plus service or with a Microsoft UltimateTV settop box.

ABC's enhanced TV broadcasts allow users to view supplemental information such as player statistics for football games, answer questions for game shows, and answer polling questions for talk and news shows (ABC 2003). The interaction takes place on a computer displaying synchronized Webcast data that corresponds to events on the TV show. The current

implementation can make it difficult for users, as their attention is needed on two screens simultaneously. In addition, the *lean forward* model of computer use is not completely appropriate for the more *lean back* task of watching TV.

Both the Microsoft/CBS and the ABC products combine Internet content with TV shows. However, neither allows users much freedom to explore. The Internet content is packaged and sent to users by the same people who created the TV program. Also, neither product personalizes either the TV show or the Internet content for individual users.

Another concept called "HyperSoap" (Dakss et al.) allows TV viewers using a special remote control to point at clothing, props and other furnishings on a soap opera in order to learn how they can be purchased. The research group studied how people interact with hyperlinked video and employed this information in developing different modes of interaction. The design of the system matches current TV viewing in that it allows users to interact with a remote control. However, one clear challenge for this model is how to deal with objects that jump around on the screen as the story jumps from cut to cut.

Jiang and Elmagarmed have introduced a novel video data model called "Logical Hypervideo Data Model" (Jiang et al. 1998). The model is capable of representing multilevel video abstractions with video entities that users are interested in (defined as hot objects) and their semantic associations with other logical video abstractions, including hot objects themselves. The semantic associations are modeled as video hyperlinks and video data with such property are called hypervideo. Video hyperlinks provide a flexible and effective way of browsing video data. However, in this system, all the associations are derived manually. Users communicate with the system using a query language. This method of interaction allows them to explore information, but conflicts with the *lean back* model of TV viewing.

Broadcast news analysis and retrieval for various purposes has also been an active area of research for a number of years. We created an initial "Personal News Retrieval System" in 1996 to test the feasibility of video broadcast filtering in the news domain (Elenbaas et al. 1999). The news broadcasts from different channels were semi-automatically indexed on a server. A client application invoked from a Web browser allows users to search individual stories. Searching is based on anchorperson, broadcaster, category, location, top-stories and keywords. □

Merlino et al. developed the "Broadcast News Editor /Navigator" (BNE/BNN) (Merlino et al. 1997). They rely on the format of the broadcast to be broken down into series of *states,* such as start of broadcast, advertising, new story, and end of broadcast. They use multi-source cues such as text cues ("back to you in New York"), audio silence to find

commercials, and visual cues such as black frame and single and double booth anchor recognition.

Hanjalic and his colleagues describe a semi-automatic news analysis method based on pre-selection of categories (Hanjalic et al 1999). They find anchorperson shots, using a template for matching the shots by matching individual frames. Also, they incorporated a simple word-spotting algorithm to form reports and use this for topic specification. Other systems have been reported in the literature dealing with the news retrieval (Ahanger et al. 1997, Brown et al. 1995, Chen et al. 1997, Maybury 2000). In addition, there is very recent research that performs automated segmentation of news and user modeling to generate personalcasts (Maybury et al., this volume).

Broadcast TV companies have also tried to come up with Internet versions of their content. For example, CNN has a limited number of current stories and an archive of old ones available in Real-video or MPEG-4 (netshow) format. (See http://www.cnn.com/videoselect/ for more details.)

The difference between our applications MyInfo and InfoSip and the cited systems is threefold: (i) our applications integrate both Web and TV content, as opposed limiting users to a single source, (ii) our interface employs a TV-like interaction, and (iii) MyInfo performs extensive prioritization and personalization based on detailed user preferences.

3. PILOT APPLICATIONS

In order to explore and demonstrate the usefulness of content augmentation, we applied a selective process of filtering initial ideas and concepts. In this section, we present our process and the pilot applications.

MyInfo and InfoSip are both designed to enhance the features of a Personal Video Recorder (PVR) such as a TiVo, ReplayTV, or UltimateTV. These hard disk-based settop boxes currently allow users to easily store large numbers of shows. The segmented news stories, movies and supplemental information from the Web will all be stored on a PVR for access by users using a traditional remote control that has a few additional buttons. These applications are not currently intended to work with live broadcasts.

3.1 The Design Process

We began by conducting a brainstorming session that included engineers and designers with experience in video processing, Web information

retrieval, and Web and interactive TV design. We produced twenty concepts that coalesced into the following themes:

- Connect: Connect users with each other, with their community; with the live world.
- Explore: Support users' ability to move deeper into a specific topic. Allow users to specify the level of detail they require.
- Anticipate: Extract, classify, and summarize information before users request it.
- Summarize: Reduce overwhelming amounts of content (especially redundant content) into appropriate chunks based on user context.

After concept generation, we conducted two focus group sessions. Our focus group consisted of four men and four women living in the suburbs near New York City. They came from different educational, ethnic, and socio-economic backgrounds; however, they all enjoyed watching TV and all had access to and experience with using the Web.

Our first session focused on evaluating and prioritizing the different concepts. In addition, participants shared their current strategies, preferences, and gripes for watching TV and collecting information from the Web. The following two concepts received particularly high ratings from participants:

1. Personal News: the application supplements TV news stories with richer detail obtained from the Web.
2. Actor Info: the application displays Web links for actors in the movie currently being viewed.

Our second focus group employed the same participants, and used a participatory design approach to better define the pilot applications. In exploring the personal news concept, participants revealed that they currently sought out news using a niche surfing technique. When they wanted to know something like the price of a stock, the outcome of a sporting event or the weather, they would tune their TVs to an appropriate channel such as ESPN (sports), MSNBC (finance), or the Weather Channel and then wait for the information to appear. They generally did not use the Web for this sort of high-level news because it required them to abandon household tasks such as making breakfast or folding laundry in order to go upstairs and boot a computer. They desired a system that offered faster access to personal news around the themes of sports, finance, traffic, weather, local events, and headlines. They wanted access to the *freshest* information for these *content zones* from any TV in their home.

In exploring the Actor Info application, participants really liked the idea of viewing supplemental information for a movie, but they did not want to

be interrupted. Instead they wanted to be able to easily ask questions such as: Who's that actor? What's that song? Where are they? What kind of shoes are those? etc. They wanted the answers to these questions to appear immediately on the screen in an overlay. This way, they could get the information they wanted without interruption. They did not want links to Web sites. Instead, they wanted much more digested and summarized information. For more detail on the design process, please see (Zimmerman et al., this volume).

3.2 MyInfo

Users access the MyInfo application via a remote control. They can select any of the six content zones identified by the focus group in order to see personal Web extracted data and the latest TV stories that match this zone. In addition, users can press a button labeled 'MyInfo' in order to see a personalized TV news broadcast that displays TV news and Web extracted data from all of the content zones.

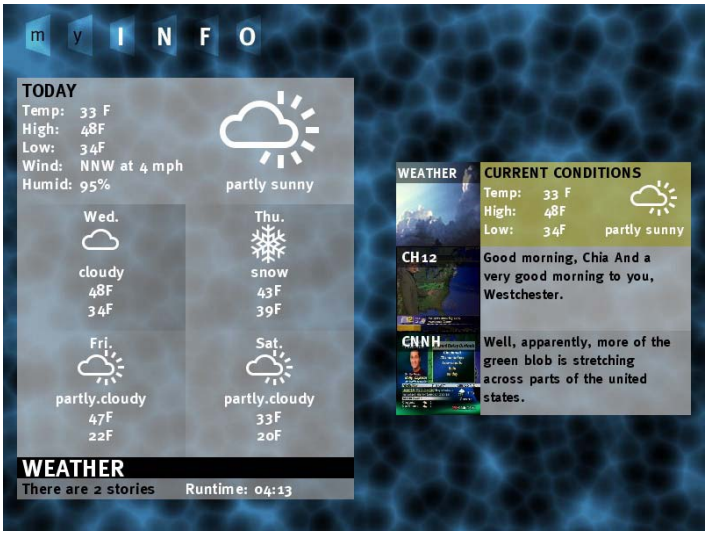

Figure 5-3. Weather screen with Web story highlighted.

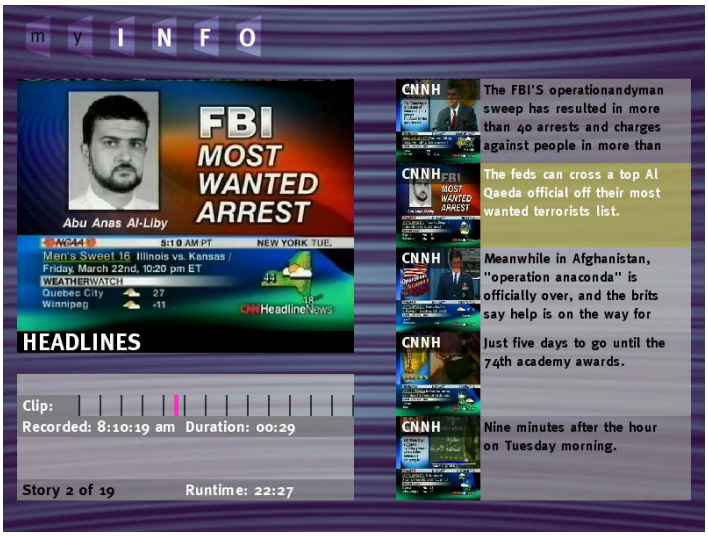

Figure 5-4. Headlines screen with TV story highlighted.

The interface displays an expanded story on the left, and a prioritized list of stories on the right. The top story always contains the Web-extracted information, which matches specific request in the user profile. The Web-extracted information includes: for weather, a four-day forecast for the specified zip code; for sports, the latest scores and upcoming games for specified teams; for local events, a prioritized listing by how soon the event will happen, distance from the home, and degree of match to keywords in profile; for traffic, delays for user-specified routes and "hot-spots"; and for finance, current prices for stocks, change in price, and percent change for indexes, stocks, and funds listed in the profile.

By pressing the NEXT button, users can navigate down in the list of stories. This allows them to effectively skip stories they do not want to hear. In addition, they can press the PLAY-ALL button in order to automatically play all the stories in a single content zone. The interaction supports users' lifestyles, and takes a step towards a *lean-natural* interface. Users can quickly check information such as weather and traffic right before they leave their homes. They can also play back all, or sections of, the personalized news as a TV show, leaving themselves free to carry out tasks in their homes such as eating, cooking, and laundry.

3.3 InfoSip

The InfoSip pilot application allows users to *sip* information about actors in a scene while watching a movie. Users press the WHO button on the

remote control and detailed information appears at the bottom of the screen. Currently, our system provides an image, a biography, a filmography, and current rumors, for all actors in the current scene (Figure 5-5). We manually extract the image from the video, but we hope to automate this process using our actor identification algorithms (Section 5.5). The descriptive information is automatically extracted from the Web. This application has an advantage over supplemental metadata supplied on DVDs, in that it is always up to date. In the example below, Tim Robbins' filmography details work he did in 2002, even though the source movie, Robert Altman's *The Player*, was released in 1992.

Figure 5-5. InfoSip screen.

During the collaborative design session, the participants stated that they often saw an actor whom they recognized but could not place. They wanted a simple method of selecting one of the actors, and seeing enough information to help them remember where they had seen that actor before. The decision to display all of the actors in the current scene takes a step towards a *lean-natural* interface by allowing users to both *sip* the metadata and view the movie simultaneously. Listing all actors in the movie would generate too large a list to navigate and would run the risk of drawing the user away from watching the movie. Displaying only the actors currently on screen would often require users to scan back in the movie, because, by the time they realized they wanted the information and grabbed the remote control, the shot with the actor they wanted might have ended. The filmographies have two pieces of additional information that support

functionality that was designed but not yet implemented. Their display can be personalized by using a viewing history to highlight movies the user has seen the specific actor in, aiding the recognition task. In addition, when filmographies contain movies that match movies scheduled for broadcast, users can use this interface to select movies for recording.

3.4 Demonstration

We developed these applications to stimulate conversations between stakeholders in the TV/Web content value chain, from media producers, packagers, distributors to media consumers. The original idea was to develop these applications as demonstrators in order to explore the target applications for consumers. We hoped to use the applications to generate business models and new application concepts with colleagues in the content creation, broadcasting, and distribution domains. However, in the future, we plan to perform a qualitative evaluation of these applications with users.

4. SYSTEM OVERVIEW

The system diagram in Figure 5-6 shows the high-level chain of content processing and augmentation. Unannotated or partially annotated content is delivered to the service provider (e.g. content provider, broadcaster) where generic analysis and augmentation is performed.

Content and (optionally) metadata are delivered to the first step (Feature Extraction and Integration) of the processing chain. At the *server* stage of the augmentation, the system extracts features and summarizes the content, generating descriptive metadata. (A more detailed description of this step is given in Section 5.) The generated metadata, in conjunction with any existing metadata, is then used to augment the content with additional information from Web sources. This information is provided by using Information Extraction from Web pages (WebIE), as described in Section 6. The augmentation (Augmentation) that occurs at the server side is general, in that it is not based on any personal profile. Following broadcaster augmentation, the content with the complete metadata is formatted and delivered to the consumer device (Formatting).

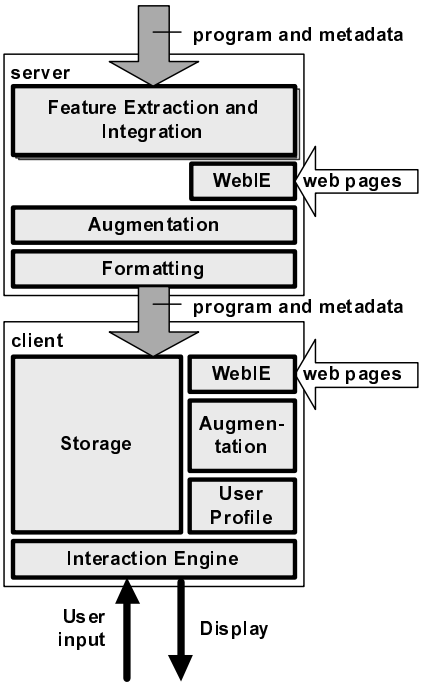

Figure 5-6. Content Augmentation system diagram

The remaining augmentation is performed in the *client* stage. Here, a consumer device has the capability of storing content, metadata (in Storage), and user profile (User Profile). The device also has a prioritization module that relies on the user profile. This is used to perform a secondary augmentation (Augmentation) with Web information (WebIE), but this time based solely on user preferences. The information obtained is stored together with the content and is presented to users (Interaction Engine) as if it were a part of the original program. One of the reasons we kept all personalization on the client was to help insure privacy, a major concern of users in our focus group.

There are several delivery pathways for the augmentation data, depending on the implementation of the system and the business model. Encoding metadata with the media is the most straightforward approach to delivering augmentation, but alternative pathways are also possible. Web broadcasts or subscription-based data retrieval can also offer localized or personalized versions of the augmentation data. Finally, the principle division in the server and client stage in Figure 5-6 is mainly to emphasize

various aspects of the system. Implementations of the system where various client functions are provided by the server, and, inversely, server functions performed by the client, are possible.

5. CONTENT PROCESSING

Methods for automatic metadata extraction can be divided into coarse- and fine-grain segmentation and abstraction. In this section, we briefly introduce the methods used for our applications. For MyInfo, we coarsely segment the news broadcast into individual stories as described in Section 5.1. Next, each story is summarized by a representative textual summary and a frame that captures the visual summary. Text summarization is described in Section 5.2. Visual summarization is performed by detection shots of the news anchor (as described in Section 5.3) and selection of the most important visual key element (as described in Section 5.4.) For InfoSip, we apply person identification using both face and voice identification, as described in Section 5.5.

5.1 Coarse Segmentation

Our approach exploits well-known, previously reported, cues to segment commercials and news segments from news programs (Merlino et al. 1997 and Boykin et al. 1999). We first find the commercial breaks in a particular news program, and then we perform story segmentation within the news portion. For stories, we use the story break markup (“>>>”) in the closed captioning. In addition, we have investigated the detection of story segment boundaries at a macrosegment level (McGee et al. 1999, Dimitrova et al. 2003).

There is a variety of commercial detectors that perform text, audio, and visual analysis to determine if TV programs contain commercial breaks (Blum 1992, Bonner et al. 1982, Boykin et al. 1999, Merlino et al. 1997). Since our domain consists of “commercial aware” programs, in which the anchors announce that a commercial break is coming up, we were able to use a computationally inexpensive, genre-specific, text-based commercial detector. In part, this relies on the absence of closed captioning for 30 seconds or more, and in part, it relies on the news anchors using cue phrases to segue to/from the commercials, such as, “coming up after the break” and “welcome back”. We look for onset cues such as “right back”, “come back”, “up next” and “when we return”, in conjunction with offset cues, such as “welcome back” and the “new speaker” markup (“>>”). We tested commercial detection on US broadcast of four financial news and four talk

show programs totaling 360 minutes, with 33 commercials totaling 102 minutes. The financial news programs included four half hour shows of CNN, NBC, and public television programs. The talk shows included four one hour late night shows on the NBC and ABC TV stations. Our algorithm detected 32 commercials totaling 104 minutes. Of these, 25 were exactly right. Only one commercial was completely missed. We detected 4 extra minutes spread out over seven commercials. The resulting recall and precision are 98% and 96% respectively.

5.2 Text Summarization

Each broadcast news story has to be summarized, in order to use (i) the abstracted data, for matching against the personal profile, and (ii) the summary, for presentation browsing. For MyInfo, a summary consists of a sentence of text and a representative image (key frame), plus a categorization (an assignment of the story to one of our six "content zones").

The summarization process begins with collection of the closed captioning text – the transcript of the spoken text - sent with each frame of the story. Figure 5-7 presents one such time-stamped transcript.

```
5252  >>> JURORS WILL RESUME
5282  DELIBERATIONS THIS MORNING
5322  IN A TWO-DECADE-OLD MURDER CASE.
5374  NORMAN REED FACES 25 YEARS
5424  FOR THE EXECUTION STYLE MURDER
5473  OF GREENBURG BOOKMAKER,
5513  RUDY WILLIAMS.
5556  HE WAS KILLED BACK IN 1979, BUT
5602  AUTHORITIES FINALLY MADE A BREAK
5658  IN THE CASE LAST YEAR.
5707  POLICE SAY REED AND THREE OTHERS
5744  WENT TO WILLIAMS' HOME
5781  TO STEAL DRUGS AND CASH,
5826  BUT WOUND UP SHOOTING HIM
5866  AND HIS STEPSON INSTEAD.
5961  >> WHEN YOU DON'T KNOW WHO,
6011  YOU KNOW, OR YOU DON'T KNOW WHY,
6100  THEN YOU WONDER IF THIS IS
6192  NOT GONNA COME TO YOU,
6235  OR IS IT GONNA DAMAGE YOU
6274  AND DAMAGE YOUR FAMILY?
6324  SO WE'RE LIVIN' IN FEAR.
6383  >> THE DEFENSE MAINTAINS REED
6419  WAS AT THE SCENE FOR A DRUG DEAL
6459  AND TOOK OFF WHEN THE OTHERS
6505  INVOLVED STARTED SHOOTING.
```

Figure 5-7. Time-stamped transcript of a news story, as collected from the closed captioning. ('>>' indicates 'change of speaker'.)

While this text could be in mixed upper/lower case, just as the sentence you are reading right now is, in practice, it is very commonly mono-case. So recapitalization is performed: the text is put entirely in lower case, and selected words are then capitalized, based on:

- Sentence-terminal punctuation (so the first word of the next sentence will be capitalized)
- Lists of:
  - First names of people
  - People name "particles" (e.g., von, del, ben)
  - Titles and honorifics ("Judge", "Senator", "Esquire")
  - Names of places:
    - Geographic regions (rivers, mountains, etc)
    - Political entities (cities, counties, states, provinces, countries, etc)
    - Terms used to denote streets, squares, bridges, parks, etc
  - Acronyms
- Simple heuristics governing capitalization of words preceding or following words in these lists. These produce, e.g., "Brooklyn Bridge", rather than "Brooklyn bridge", and "George Washington" rather than "George washington".

Figure 5-8 shows the result after the recapitalization step.

Jurors will resume deliberations this morning in a two-decade-old murder case. Norman Reed faces 25 years for the execution style murder of Greenburg bookmaker, Rudy Williams. He was killed back in 1979, but authorities finally made a break in the case last year. Police say Reed And three others went to Williams' home to steal drugs and cash, but wound up shooting him and his stepson instead. >> When you don't know who, you know, or you don't know why, then you wonder if this is not gonna come to you, or is it gonna damage you and damage your family? So we're livin' in fear. >> The defense maintains Reed Was at the scene for a drug deal and took off when the others involved started shooting.

Figure 5-8. The recapitalized text

Our own algorithm was used, which was adequate for our needs, but not perfect. The reader should note the mistaken capitalization of 'and' and 'was', because the surname 'Reed', being also a common first name, is in the first names list, and a heuristic that texts always contain persons' full names fired. Better algorithms have been developed (Brown et al, 2002) which first capitalizes the whole text, and then de-capitalizes those words in a list of common English words.

The IBM INTELLIGENT MINER FOR TEXT ("TextMiner") document summarizer (Boguraev et al., 2000) is then applied, to select the N sentences in the story which summarize it best. (We use N = 1.) "Best" in this context is a weighted metric, involving the "salience" (position) of the sentence in the document, its length, and other factors. Usually, the first sentence of a news story ends up being the one selected; given how news stories are written, this sentence is normally both a comprehensive summary and a good introduction to the story. However, sometimes a non-useful sentence occurs first ("Hello, I'm Dan Rather."); TextMiner often catches these cases and makes a better selection.

For the story in Figure 5-8, the text summarization found is:

"*Jurors will resume deliberations this morning in a two-decade-old murder case.*"

We also use the TextMiner document classifier. It works on the basis of frequency of occurrence of words within the story, and similarity of such frequency distributions to canonical examples. The classifier engine is domain-independent; to use it, we trained it off-line with a corpus of exemplar stories for our six "information zones".

In the case of the story in Figure 5-8, the computed categorization is:

15.2041 headlines
12.8685 future announcements ("teasers")
11.6219 commercials
11.4269 local news

where the numbers on the left are confidence scores, which have no metric interpretation; simply, larger values are to be preferred to smaller values.

A third TextMiner engine, the "feature finder", is used to extract proper names from the story. These names could be matched against entries in the user profile, to determine the story's relevance for the user.

Name, person: Norman Reed 4708 25
Name, unknown: Greenburg 4820 13
Name, person: Rudy Williams 4847 42
Name, person: Reed Was 5743 15

Here, the numbers to the right of the names of persons, places, and unknown things are the locations and durations of their occurrences in the story (in units of characters). The number of occurrences of a given name, and its salience in the story, could further contribute to calculation of a story's relevance for the user.

The TextMiner classifier was evaluated as part of the NIST Tipster SUMMAC text summarization evaluation of 1998 (Mani et al. 1998). It was found to have a precision of 0.68 and a recall of 0.47.

5.3 Anchor Detection

We have an anchorperson detection module, which is an important contributor to multi-modal segmentation, because stories often begin and/or end with in-studio (anchorperson-present) shots, rather than during *reportage* segments (shots of reporters and/or interviewees, commonly taken "on location", or at the reporter's "desk"). This module is also important for story summarization, since it helps in choosing representative keyframes for the stories that do not include anchor images. This module is composed of three main blocks:

1. Shot detection
2. Face clip finding
3. Anchorperson shot detection

5.3.1 Shot Detection

We compute the cumulative probability distribution of each of the red, green, and blue channels from their histograms for each frame (Hampapur, et. al., 1994). The distance between two cumulative probability distributions is found by using the Kolmogorov-Smirnov (KS) test. First- to fourth-order differences, and two types of tests, are used to make shot boundary detection robust with respect to various video effects (wipes / fades / dissolves) and flashes.

We compute KS distances between consecutive frames and between those separated by 1, 2, and 3 frames. The first test is based on ratio combinations of each of these distances. Each of the ratios (and/or the distances) must be larger than predetermined thresholds. In order to prevent false shot break detections due to flashes, we also check the KS distance of the following frame to the previous frames before declaring the current frame as the starting point of a new shot. Second- and third-order differences, and different thresholds on the distances and their ratios, are used for this. Again, the ratios (and/or the distances) must be larger than thresholds. For example let $D(n, n-2)$ denote the KS distance between frames n and $n-2$. If

$D(n, n-2) / D(n-2, n-3) >= thr1$ *and*
$D(n, n-3) / D(n-3, n-4) >= thr2$ *and*
$D(n, n-3) > thr3$ *and* $D(n, n-2) > thr4$,

one of the two acceptable conditions is satisfied and the same test is applied for the following frame (by replacing *n* with *n+1*) before declaring the current frame (frame *n*) as the beginning of a shot. The same threshold values are used for all videos.

5.3.2 Face Clip Finding

We process the video sequence to find video clips containing faces. For each clip, the following information is saved: the frame numbers of the first and last frames of the clip, and the coordinates of the bounding rectangle of the largest face in view, in each frame in the clip. These clips are, in general, subsets of the shots found in step 5.3.1. In cases where the clip spans two shots; it is broken into two clips, at the shot boundary. Where multiple video clips containing faces occur within a single shot, they are merged. We use a face detection algorithm which is based on flesh tone-finding followed by high chroma detection and horizontal texture detection (Connell 2002).

5.3.3 Anchorperson Shot Detection

The next step is matching the face in one clip with those in others. For each face clip, the largest face (as determined by its rectangular bounding box) is used. To make the matching more robust with respect to head motions, especially roll (rotation in the image plane), we expand the original bounding box so that it is twice as large in both width and height, in order to include the hair region of the head and some of the shoulder region, also. Thus, more color information is incorporated, in addition to just flesh tone. The enlarged rectangle is then divided into 6 regions: one for each side of the head, one for the original face box and the hair on the top, and three for the shoulder and neck region. See Figure 5-9.

For each frame, the cumulative distributions of the red, green, and blue channels are calculated for each of the six regions; these are then averaged over all the frames of the clip. This gives us a representative distribution for that clip. Although this is quite a slow process, non-real time performance can be tolerated in the MyInfo application.

To find the canonical anchorperson clip, we look for one clip whose representative distribution is very similar to those of a large percentage of the other clips. We compute the pairwise KS distance between the representative distributions of each pair of clips, and, if the distance is less than a threshold, we consider them to be different image sequences of the same person and background. As this calculation is of order n^2 in the number

of clips, we employ an early-termination scheme to ignore some of the clips. First we start with the representative distribution of a clip and find the KS distance of this distribution to those of the other clips. If we find that *m*% of the clips have distances that are less than a threshold *t*, we group them together, and sort the list in ascending order according to their distances to the distribution we have started with. We call this group the *initial list,* and then we start pruning this group. We take the first clip we started with and the one which is at the top of the *initial list,* and put them in a new group called the *anchor shots list* which will be the pruned version of the *initial list.* To start pruning, we take the first distribution in the *initial list* and find the KS distances of the others in the list to it. We keep the ones whose distances are less than *t* in the *initial list* and remove the others and again sort the list. We then put the first element of this list in the *anchor shots list* and repeat the process. This way we make sure that every clip in the *anchor shots list* will be within distance *t* of each other.

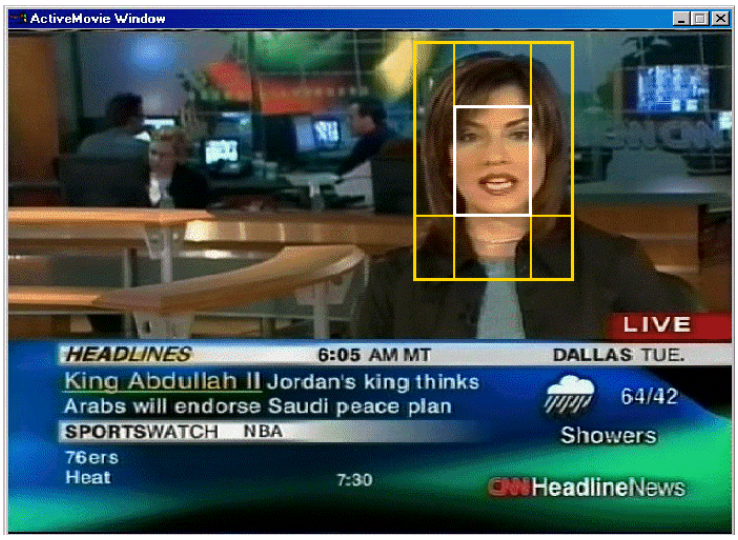

Figure 5-9. Face, with original (inside box) an expanded and partitioned (outside box) bounding box.

In our experiments, the percentage threshold *m* was 25-35%. In its current state, the algorithm can only deal with clips in which there is one anchorperson, but it can be extended to work on other more general cases also.

We performed some experiments on four news segments to test the performance of the algorithm (see Table 5-1). The algorithm was tested on 56,500 frames of news videos from CNN and local news channel in

Westchester, NY. In this data set, there were 26 true anchor shots. (Note that in the test data, there are some shots in which more than one anchorperson appears. As the current version only works for single anchorperson in a shot, the shots with multiple anchorpersons have not been counted in the true anchor shots.) The algorithm found one false positive and five false negatives. The percentage of anchor shots whose starting frame was detected correctly by the shot detection algorithm was 96.15%. If some special effect is used for transition from one shot to another, detection of beginning of the new shot is delayed. In all these experiments, the same KS distance threshold was used.

Table 5-1 Results from anchor detection.

	Number of detected shots	**Number of detected face clips**	**Number of true anchor shots**	**Number of false positives**	**Number of false negatives**	**Accuracy of the shot beginning points**
CNN1	□□	□□	7	0	1	100%
CNN2	102	68	7	0	2	100%
Ch12_1	201	63	5	1	1	100%
Ch12_2	132	99	7	0	1	85.7%

5.4 Summary Image Selection

In order to find a representative visual key element, we find a representative keyframe for each news story. This image has to be the most representative frame from the story. Figure 5.10 shows an example of the summarization process for a news story. At the top, a film strip consisting of 8 families of video frames is presented, showing the length of each family along with its cumulatively averaged histogram. For finding important segments, we use the uniformly colored segments generated by family histogram clustering; the frames are weighted by the duration of the family they belong to. We use Family Histograms (Dimitrova et al., 1999) to find uniformly colored video clips. These correspond to shots, or parts of shots.

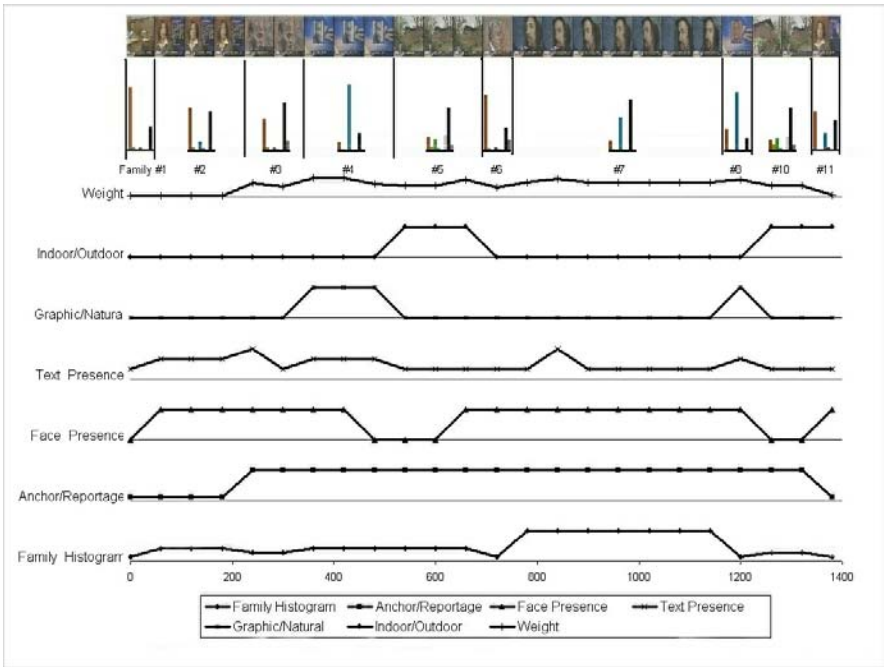

Figure 5-10. Representative Image Selection

For each feature, we find a value between zero and one that gives the "desirability" of that feature. The figure shows the various visual features that are extracted for summary image selection. For the family histograms, the importance of a frame is derived by the duration of the family it belongs to and divided by the longest family in the news story. The bottom curve shows the importance based on family histograms. The second curve shows the anchor vs. reportage class. The news story initially starts with the anchor, goes on to reportage and ends with the anchor. Each story is usually composed of an anchor shot followed by reportage shot(s). The anchor shots are similar for all stories, so they do not provide any value in representing the story. In order to select an image from only the reportage, an anchorperson detector is used (see Section 5.3.3). In this curve, the value of anchor is 0.1 and that of reportage segment is 1. The third curve from bottom shows presence or absence of faces. The value of this feature is 0 if no faces are present, and 1 if one or more faces are detected. The next curve gives the text importance in the video. This is derived by the number of lines of text in the frame divided by the maximum number of lines in the news program. Presence of both faces and text is desirable in the selected image. The next curve in the graph shows presence of graphic vs. natural scene video. Graphic information relates to shots that contain graphs, slides, and other

computer-generated screens. We include a graphic image if available. The second curve from top gives the indoor vs. outdoor information. For news programs, we feel that outdoor shots are more important for news stories than indoor shots. We use the indoor/outdoor detector developed by Naphade et al. (2002). In the above curves, the value is 1 for graphic and outdoors and 0 for natural and indoor frames. We select a frame that is deemed to be the most `interesting' by the algorithm that considers all the above attributes. An importance score is computed for each frame as following:

$$FrameScore = AR * \left(\sum_{i=1}^{6} W_i F_i \right)$$

Where AR is 1 for reportage segment and 0.1 for anchor segment. The Wi is the weight given to each of the features Fi: F_1 is Face, F_2 is videotext, F_3 is anchor or reportage, F_4 is graphics or no graphics, F_5 is outdoors or indoor, and F_6 is weight of family histogram. The top curve in Figure 5-10 gives an importance score based on all the input features for each frame. For our system, presently we use equal weights for all the features. A frame from family #3 is selected as the most representative for the story.

We performed an empirical benchmarking of our method in the following way. We found two representative images, one using our image selection algorithm, and another image using the “middle image pick method” as taking the image occurring at the middle of the story. We watched the news story and determined which image summarized the news best. Based on this viewing, we decided the “desirability” of the image selected on a scale of 1 to 5. On this scale, if the image selected was the one that we felt summarized the news story the best, we gave it a 5; in the other extreme where the image was not desirable at all, we gave it a rating of 1.

Based on this system, we analyzed a total of one hour of news stories consisting of half-hour each of CNN Headline News and Channel 12 news (local news channel). A total of 33 news stories were selected to be presented to the user for evaluation. The Table 5-2 shows the number of votes for the ratings of the middle image pick and summary image selection algorithms. Overall, the algorithm does better than the mid method. The average rating of the image selected by the algorithm is 4.27, vs. 3.87 using the mid method. Also, the standard deviation of the algorithm is only 0.87, compared to 1.17 of the mid method, which means that the algorithm consistently gives better images.

Table 5-2 Results of the middle image vs image selection method.

Rating	Mid Method votes	Algorithm result
5	14	16
4	5	12
3	12	3
2	0	2
1	2	0
Average	3.87	4.27
Std Dev	1.17	0.87

5.5 Person Identification

A rich "frequently asked question"-answering application relies on manual annotation or automatic detectors. For example, to answer the "who is this person?" question in a movie, documentary or home video, we need to know which people are present in each scene. The major challenge is to robustly identify persons from different views, distances, lighting conditions, in the presence of various background noise conditions. We used automatic face and voice identification methods for this task (Li et al. 2001).

A person identification approach is constructed, based on the joint use of visual and audio information. First, in the *analysis* phase, we perform visual analysis for detection, tracking and recognition of faces in video. Face trajectories are first extracted and the Eigenface method is used to label each face trajectory as one of the known persons in the database. Due to the limitation of existing face recognition techniques and the complex environmental factors in our experimental data, the visual recognition accuracy is not high. Next we employ audio segmentation and classification to find the speech segments. Film often has music background or environmental noise in the soundtrack, and this factor makes the audio identification a challenging process. Speaker identification using Gaussian Mixture Models is applied to the speech segments. Both audio and visual analysis have their advantages under different circumstances, and we studied how to exploit the interaction between them for improved performance.

In the fusion phase, two strategies have been employed (Li et al., 2001). In the first strategy, the *audio-verify-visual(AVV) fusion* strategy, speaker identification is used to verify the face recognition result. The second strategy, the *visual-aid-audio fusion (VAA)* strategy, consists of using face recognition and tracking to supplement speaker identification results. In our

testing we used a database, which consisted of 100 video clips (dialog, non-dialog, and silent clips) from the sitcom "Seinfeld." In the experiment, speaker identification gave recall of 54.6%, and precision of 76.9%, while the face recognition gave recall of 15%, and precision of 35%. The AVV strategy yielded 12% recall, and 92.9% precision, while the VAA strategy yielded 62.9% recall and 82.4%. precision. We see that AVV has a slightly lower recall than the face recognition and best precision which is good for surveillance type of applications. VAA generates the best overall identification performance and is suitable to TV content analysis applications such as InfoSip.

In addition, we use textual information extracted from closed caption or video caption. We have a name spotting process that extracts role names that appear in each video scene, and assigns a score for each detected role name according, to the frequency of its own appearance as well as that of those that closely relate to it. These scores, together with our audiovisual detection results, are used in a final voting process to decide which role(s) appear in the scene. The integration is based upon the belief values of different candidates, using a single layer Bayesian network. The ones with highest integration belief will then be justified as top characters appearing in the scene.

For narrative content where there is more than one talking face on the screen each time, and sometimes non-related voice over, we need to use a talking head detection process, which automatically detects the face(s) on the screen that has corresponding speech in the synchronized soundtrack. Such information can then be used in the fusion process to integrate the speaker identification results with the corresponding face trajectory. A cross-modal association method called Cross-modal Factor Analysis (CFA) is proposed and used for our talking head detection (Li et al., 2003). CFA achieves 91.1% detection precision in our experiments, while our two other implementations based on Latent Semantic Indexing (LSI) and Canonical Correlation Analysis (CCA) achieve 66.1% and 73.9% detection precision respectively using the same set of testing data.

6. WEB INFORMATION EXTRACTION

Unlike in-depth Natural Language Processing, Information Extraction (IE) "skims" the input text, finds relevant sections and then focuses only on those sections in the subsequent processing in order to find targeted information (Cardie 1997). In other words, IE systems (1) take as input a

document that contains unrestricted text, (2) find useful information about the domain from the analyzed text, and (3) encode the information in a structured form (e.g. suitable for populating databases). We will refer to IE in the context where the input is a Web document as Web Information Extraction (WebIE). An introduction to IE, WebIE, and additional references are given in (Janevski, 2000).

Our system implements a framework in which instantiations of modules called *IE rules* can be plugged in and executed for each acquired document. We developed two collections of rules: tag-based and content-based. Tag-based rules utilize the encoding of the documents (tags), while content-based rules apply natural language processing techniques over the text and operate at various levels starting from keyword matching to in-depth syntax analysis. We will refer to IE rule instantiations as *IE tasks*.

6.1 Laser WebIE

We distinguish two types of WebIE – Diffusion and Laser. In Diffusion WebIE, tasks require broad search over a large number of sites and time is not critical. A Laser WebIE system extracts and formats information from a well-defined set of Web sources. Our content augmentation system executes instantiations of Laser WebIE rules that retrieve information on news headlines, weather, traffic, sports games and scores, stock quotes, and movie cast information. Furthermore, most IE tasks are customized for every instantiation. Specifically, the weather information is tied to the user's zip code; traffic information is dependent on the user's route to work; sports and stock depend on user's personal preferences; movie cast information depends on the cast member currently present in the scene. We will use the segment of the user personal profile in Table 5- 3 to illustrate the WebIE tasks in this section.

Table 5- 3. Part of user profile - a sample

Zip code	10510
Traffic hotspots	Taconic Pkwy; Bear Mountain Bridge; Route 100; Tappan Zee Bridge
Stock symbols	PHG; IBM
Favorite Actors	Bening, Annette; Spacey, Kevin; Redford, Robert

6.2 Document acquisition and IE rules

Once WebIE task(s) are instantiated, the results must be delivered quickly while the video context is still active. Even with high-speed access to the Web, it could take considerable time to retrieve, process, and present

information to the user. For this purpose, the source URLs are given in advance and WebIE tasks directly acquire the Web pages, thus avoiding a search through numerous pages. To bootstrap the augmentation, a list of predefined URLs for each of the queries is embedded in the system. Since content augmentation is likely to be delivered as a service, content creators and/or distributors can encode these pointers with the content, or have them delivered to the system ahead of the broadcast delivery of the content (or during delivery). Moreover, in a scenario where all content processing is performed locally, "generic" URLs (pointers) would provide enhancements for various WebIE tasks. The URL for the information source is given partially, and is then customized based on the WebIE task arguments and the information from the personal profile. An example local weather URL is given in Table 5- 4a) where at least part of the URL (in bold) is dynamically generated. Another example is extraction of actor information in Table 5- 4b) using the generic URL and customizing it with the actor name Robert Redford (in bold).

Table 5- 4. WebIE: weather and actor information extraction - a sample

a)	http://weather.com/weather/local/**<zipcode>** http://www.weather.com/weather/local/**10510**
b)	http://www.imdb.com/Name?**<last>,+<first>** http://www.imdb.com/Name?**Redford,+Robert**

For the design of Laser WebIE tasks, we assumed a relatively static content presentation style since Web site structures remain stable for a period of time. The IE tasks take advantage of this and use identifiable references specific to the information source. However, the number and the uniqueness of each source of information argues against the desire to build as few WebIE rules that can instantiate as many tasks as possible. All IE rules are built on the same principle, and use a similar set of parameters to identify an IE rule. First, the boundaries of a segment are specified. Second, the boundaries of the extracted information are given. And, third, the format of the output data is defined. The segments and the extracted information can be defined through HTML tags or specific contents such as keywords, numbers, dates, and other data types. In addition, IE rules can take advantage of the segment structure (e.g. tabular information representation), and use it to identify a segment and/or the information that needs to be extracted.

In Table 5- 5, we show two IE task examples with the corresponding URLs, the segments and the extracted information. The Stocks task will acquire a document from a URL that contains the stock quote for Philips

(PHG) – the customized part is given in bold. Then, the segment is isolated, based on specific HTML tags also given in bold. Finally, the task extracts texts from two such regions shown in gray background. The extracted texts contain the current stock price, the absolute and the relative change in value. The execution of Headlines, also shown in Table 5- 5, will access a URL customized with the fragment in bold – the number 11 stands for Westchester County. The segment is isolated based on two keyword phrases provided in the task definition. The result is extracted between the two characteristic HTML tags. "White House press briefings", one of the extracted headlines, is shown on a gray background. For each extracted headline the task will also return the URL of the document that contains the complete story – all segment tasks look for links within the extracted region, and if one is found, the URL is returned with the result.

Table 5- 5. Information extraction from stocks and headlines – a sample.

Stocks	URL	http://qs.money.cnn.com/apps/stockquote?symbols=**phg**
	Segment	<td … class="stockheader">31.13</td><td …>
	Extracted text	{<td …>31.13</td>, <td …><img …>0.90 / +2.98%</td>}
Headlines	URL	http://www.news12.com/CDA/0,2033,**11**,00.html
	Segment	What You Need To Know … White House press briefings … National & International News
	Extracted text	White House press briefings
	Extracted URL	/CDA/Articles/View/0,2049,11-11-22511-258,00.html

While all WebIE task examples show Web pages written in English, in general, the WebIE rules and tasks are easily portable to other languages. For rules and tasks that are based on keywords and property of the page content, porting to another language is straightforward. In the cases where in-depth syntax analysis is performed to extract information, a larger effort would be required to integrate corresponding language processing tools, such as syntax parsers, with the system. The applicability of the system described depends heavily on the robustness and performance of the information extraction components. Once defined, Laser WebIE tasks are very accurate, as long as the structure of the source Web page(s) remains unchanged. In our tests, we ran a combination of about fifteen WebIE tasks daily for thirty days and obtained 100% accurate extracted information. Laser WebIE tasks have such high accuracy and robustness because they were defined for specific type of target Web pages. The properties of the WebIE tasks depend highly on the content delivery business model. Narrowly defined Laser WebIE tasks are suitable for a setup where a dedicated service maintains the annotation and augmentation.

7. PERSONALIZATION

Personalization provides one of the greatest benefits and one of the greatest risks to content augmentation applications. During our focus group sessions, participants constantly stressed their desire for personalized and easy to use information, along with a need to feel in control. However, they were very wary of any system that made them feel watched. They were all quite uncomfortable with the idea of broadcasters and advertisers gaining access to their detailed information about their media consumption habits, patterns and preferences. Our approach to the personalization challenge was to use our focus group to help identify areas of greatest benefit, and then to balance this with technological capabilities and privacy protection. Based on these requirements we designed the MyInfo application to personalize news in two ways. For the Web data, the system parses and extracts information from Web sites according to requests in the user profile. For TV news stories, the application prioritizes individual stories based on time of broadcast (freshness of this news), topics of interest listed in the user profile, and cues broadcasters use to indicate a story's importance.

7.1 Personalizing Web Data

In discussions with our focus group, participants stressed that they did not want to spend a lot of time configuring their system in order to get personalized information. They claimed that difficulty in setup (or perceived difficulties) as well as the requirement to share personal information kept them from using current Web news personalization systems like myYahoo (www.myyahoo.com). Therefore our system places all of the personalization in the client device (settop box in the home) and focuses on providing maximum, targeted information with minimal input. We present screen shots of expanded Web stories for financial news, traffic, local events, and sports in Figure 5-11.

The weather information and sports allow minimal interaction by using the zip code data users enter into their settop boxes while configuring their channel lineups. MyInfo automatically extracts the weather information for this zip and extracts the latest sports scores and upcoming games for local teams. If users desire, they can edit their profile and request weather for a different zip and select other sports teams to track.

Financial news and traffic require more input from the user, but the resulting feedback makes the effort worthwhile. For traffic, the profile contains a set destinations and a set of "hot spots". Destinations include

towns or prominent structures such as malls, stadiums, airports, etc. Hot spots include points of constriction like bridges and tunnels, which notoriously have traffic delays. Once selected, the system extracts Web traffic information on the specific hot spots and on the major roads between the users home and the selected destinations. For financial news, the profile must contain a list of the stocks, mutual funds, and financial indexes the user wishes to track. The system then displays a listing of the item, its current price, change in price, and percent change.

For local events the profile contains a set of keywords describing events users like most such as “music, jazz, fairs, plays, theatre…”. The system displays a prioritized listing of these events based on how soon they will take place, their distance from the user’s home, and the match to the keywords.

Figure 5-11. Expanded Web stories for financial news, traffic, local events, and sports. These panels display on the left-hand side of the MyInfo application. For a view of a whole screen, see Figure 5-3.

The personalized Web information improves the traditional TV news experience in two ways. First, it reduces the amount of time required to retrieve this information from either a traditional TV or Web site. For

example, if users just want to know the current temperature or a stock price, the information is a single button push away. They don't even have to wait for the news anchor to tell them and they don't have to type in a URL and then enter their zip code. Second, the Web-extracted data adds personalization to the TV experience. For the first time, the TV can immediately provide users with specified information on demand. For example, the local TV news can only afford to devote so many minutes of broadcast time each day for traffic information. This prevents them from relaying information on all routes during a traffic segment; forcing them to often skip routes that are important to an individual user. The personalized Web data creates a more personal and meaningful experience, while still allowing users to also view the traditional TV traffic news, which provides a nice overview of the whole traffic situation and information on the worst spots in their area. The personalization of this information helps generate the new lean-natural experience.

7.2 TV News Personalization

MyInfo personalizes TV news stories through segmentation, classification, and prioritization. Segmentation cuts the TV news into individual stories and classification places each story into one of the six content zones. These processes allow users to manually personalize the TV news by allowing them to quickly select and skip individual stories, a big improvement over the traditional TV news viewing experience. Prioritization takes this a step further by organizing individual stories within a content zone.

In prioritizing stories, the system balances topics specified in the profile, time sensitivity, and cues the broadcaster uses to indicate a story's importance. Different formulas are used for the different content zones (See Table 5- 6).

Table 5- 6. Metadata sample

Zone	Profile Match	Broadcaster Importance	Time Sensitivity	Time Sensitivity Rule
Traffic, Sports, Financial News, Weather	40%	50%	10%	Time since or until event
Local Events	60%	20%	20%	Time until event
Headlines	40	50	10%	Length of time since/until event

Use of the broadcaster information is very important, particularly for the headlines zone. Users have no way of predicting every kind of news story that might be important to them. They may know they are interested in China, and therefore add this topic to their profile. However, it is hard for them to predict major events that affect many people, such as earthquakes, gas leaks, trial outcomes, etc. By allowing the broadcasters' editorial content decisions to play a role, users get a much better mix of information.

MyInfo determines broadcaster importance of a story from three different characteristics: (i) duration, (ii) location in the newscast, and (iii) teaser announcing a story will play later in the broadcast. Since broadcast time is limited, a longer story will be more important. Location in the broadcast and use of a teaser are subtler. The most important TV news stories generally appear at the beginning; however, broadcasters place other stories they think many viewers want to see at the end. Then they use teasers to keep the viewers from switching channels. At this time, we have designed the broadcaster importance method but it has yet to be implemented and evaluated. Currently our prototype only considers the profile in prioritizing the TV news stories.

7.3 InfoSip: Personalized/Augmented Narrative

InfoSip is an example of a "frequently asked questions" answering application. It unobtrusively serves actor information related to the scene. During focus group testing, participants indicated that they wanted supplementary information for movies and TV shows, but they did not want it to interrupt viewing. With our system, users interact by selecting a specific query on the remote control. InfoSip uses predefined categories of questions/buttons such as "who", "where", "what", "when", "why", and "how much". For example, users press the "who" button to ask "who's that actor?" The system displays a list of all of the actors in the current scene using annotated data from person identification (see section 3.3) and supplemental data about each one obtained through Web IE (Figure 5.5). Web IE allows InfoSip to improve upon supplemental information currently found on DVDs in three ways: (i) it always extracts the latest information, (ii) it can personalize information based on a user profile, and (iii) it can consult information sources other than the original content creator.

Filmography information can be personalized based on the user's viewing history. Highlighting movies in which users have seen an actor increases the chances that they will remember why this person looks familiar. The design of the menus on the overlays can also be reconfigured

based on a personal profile. For example, “bio”, “filmography”, and “rumors” are the three menus available for person interested in gossip, but “bio”, “filmography”, and “references” are menus available for people more interested in references this movie is making to other movies.

8. CONCLUSIONS

In this chapter we presented personalization aspects for content augmentation applications that combine content from multiple media sources. Our pilot applications MyInfo and InfoSip show promise that the technology has come of age. Web Information Extraction and the segmentation, indexing, and retrieval of video at a subprogram level both offer new tools for TV personalization developers. These technologies can improve the viewing experience by both better understanding the TV content and by retrieving related material that is more focused at individual users. In the future we plan to evaluate our pilot applications with real users, continue developing video and Web retrieval and extraction algorithms and generate more content augmentation concepts.

9. ACKNOWLEDGEMENTS

We thank Lesh Parameswaran, Jeanne de Bont, Henk Lamers, and Giang Vu of Philips Design for help with the user interface design of the content augmentation project.

REFERENCES

1. Ahanger G., Little, T.D.C. A System for Customized News Delivery from Video Archives, in Proceedings of ICMCS '97 (June 3-6, 1997) IEEE Press.
2. Ardissono, L., Portis, F., and Torasso, P.: Architecture of a System for the generation of personalized Electronic Program Guides. Eighth International Conference on User Modeling: Workshop on Personalization in Future TV, Sonthofen, Germany, (2001)
3. Blum, D. W., “Method and Apparatus for Identifying and Eliminating Specific Material from Video Signals, ” US patent 5,151,788, September 1992.
4. Boguraev, B. and Neff, M., Lexical Cohesion, Discourse Segmentation, and Document Summarization, Proc. RIAO International Conference, April, 2000, Paris
5. Bonner, E. L. and Faerber, N. A., “Editing system for video apparatus,” US patent 4,314,285, February 1982.
6. Boykin, S. and Merlino, A. 1999. Improving Broadcast News Segmentation Processing. IEEE International Conference on Multimedia and Computing Systems. Florence, Italy. 7-11 June 1999.

7. Boykin, S. and Merlino, A. 2000. Machine Learning of Event Segmentation for News on Demand. Communications of the ACM, 43(2): 35-41
8. Brown, M.G., Foote, J.T., Jones, G.J.F., Jones, S.K., Young, S.J. Automatic Content-Based Retrieval of Broadcast News, in Proceedings of ACM Multimedia 95, (San Francisco CA, 1995) ACM Press, 35-43.
9. Brown, E. W. and Coden, A. R., Capitalization Recovery for Text, in Coden, A. R, Brown, E. W. and Srinivasan, S (eds): Information Retrieval Techniquest for Speech Applications, Springer, 2002, pp. 11-22.
10. Brusilovsky, P. (2003) Adaptive navigation support in educational hypermedia: The role of student knowledge level and the case for meta-adaptation. *British Journal of Educational Technology*, 34 (4), 487-497.
11. Chen, L., Faudemay, P. Multi-Criteria Video Segmentation for TV news, in Proceedings of. IEEE First Workshop on Multimedia Signal Processing, Princeton, NJ, 1997.
12. Connell, J., "Face Finding", http://www.research.ibm.com/ecvg/jhc_proj/faces.html, 2002.
13. Cotter P. and Smyth, B.: PTV: Intelligent Personalized TV Guides. Seventeenth National Conference on Artificial Intelligence, Austin, TX, USA, (2000) 957-964
14. Cardie, C., "Empirical Methods in Information Extraction", *AI Magazine*, vol. 18, no. 4, pp 65-79, 1997.
15. Dakss, J., Agamanolis, S., Chalom, E., Bove, V.M., Brooks, K., Nemirovsky, P., Westner, A., HyperSoap: http://www.media.mit.edu/hypersoap.
16. Das D. and ter Horst, H.: Recommender Systems for TV. Technical Report WS-98-08 Recommender System, Papers from the 1998 Workshop, Madison, WI. Menlo Park, CA: AAAI Press, (1998) 35-36
17. Dimitrova, N., Martino, J., Agnihotri, L., Elenbaas, H., "Superhistograms for video representation," IEEE ICIP 1999, Kobe, Japan
18. Dimitrova, N., Agnihotri, L.and Jainschi, R., Temporal video boundaries, in Video Mining Book, A. Rosenfeld, D. Doermann, and D. Dementhon (eds.), Kluwer, 2003, pages 61--90.
19. Elenbaas, H., Dimitrova, N. McGee, T. PNRS - Personalized News Retrieval System, SPIE Multimedia Storage and Archiving Systems, 1999.
20. Haas, N., Bolle, R., Dimitrova, N., Janevski, A., Zimmerman, J., Personalized News Through Content Augmentation and Profiling, in Proceedings of International Conference on Image Processing 2002 (Rochester NY, September 22-25, 2002) IEEE Press.
21. Hampapur, A., Jain, R., and Weymouth, T., Digital Video Segmentation. In Proc. of the ACM International Conference on Multimedia, pages 357--364, San Francisco, 1994
22. Hanjalic, A., Lagendijk, R.L., Biemond, J. Semiautomatic news analysis, indexing and classification system based on topic preselection, SPIE Storage and Retrieval for Image and Video Databases VII, January 1999, Volume 3656, pp86-97.
23. IBM Intelligent Miner for Text™
24. Janevski, A., Dimitrova, N. Web Information Extraction for Content Augmentation, in Proceedings of ICME'02 (Lausanne, Switzerland, August 26-29) IEEE Press.
25. Janevski, A., "UniveristyIE: Extracting Information From University Web Pages", MS Thesis, U. of Kentucky, Lexington, 2000
26. Jasinschi, R., Dimitrova, N., McGee, T., Agnihotri, L., Zimmerman, J. Video Scouting: An Architecture and System for the Integration of Multimedia Information in Personal TV Applications. IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP) Salt Lake City, UT, USA, May 7-11 (2001) 1405-1408
27. Jiang, H., Elmagarmid, A. K. Spatial and Temporal Content-Based Access to Hypervideo Databases, VLDB Journal 7(4) (1998) 226-238.

28. Li, D., Wei, G., Sethi, I.K., and Dimitrova, N.: Person Identification in TV Shows, Journal on Electronic Imaging, special issue on Storage, Processing and Retrieval of Digital Media, October (2001)
29. Li, D., Dimitrova N., Li, M., and Sethi, I.K., Multimedia content processing through cross-modality association, ACM Multimedia 2003, November 2-5, Berkeley
30. Kubey, R. Csikszentmihaly, M. Television and the Quality of Life: how Viewing Shapes Everyday Experiences, Lawrence Erlbaum Associates, Hillsdale NJ, USA, 1990.
31. Maybury, M. (ed.) February 2000. News On Demand. CACM. 43(2): 33-34, 35-79.
32. Mani, I., House, D., et.al: Tipster SUMMAC Text Summarization Evaluation, Final Report, October 1998. Mitre Technical Report MTR W980000138 and Technical report, DARPA, 1998
33. McGee, T., and Dimitrova, N., Parsing TV Program Structures for Identification and Removal of Non-story Segments, SPIE Conference on Storage and Retrieval for Image and Video Databases VII (ei24) 1999.
34. Merlino, A., Morey D., Maybury, M. Broadcast navigation using story segmentation, in Proceedings of ACM MM '97, (Seattle WA, November 1997) ACM Press, 381-388.
35. ABC Enhanced TV: http://heavy.etv.go.com/etvHome/
36. Microsoft NAB demo of enhanced TV: http:// www.microsoft.com/ presspass/exec/craig/nab97.asp
37. Microsoft/CBS interactive TV: http://www.microsoft. com/presspass/press/2000/Sept00/CBSpr.asp
38. Naphade, M. R., Kozintsev I., and Huang, T.S.: A Factor Graph Framework for Semantic Video Indexing. IEEE Transactions on Circuits and Systems for Video Technology, Vol.12, No.1, (2002) 40-52
39. Zimmerman, J., Marmaropoulos, G., and van Heerden, C. Interface Design of Video Scout: A Selection, Recording, and Segmentation System for TVs. Volume 1 Proc. of Human Computer Interaction International (HCII) New Orleans, LA, USA, August 5-10, (2001) 277-281