Machine-Generated Multimedia Content

Nathan Nichols & Kristian Hammond
Intelligent Information Laboratory
Northwestern University
2133 Sheridan Rd
Evanston, IL 60208
(847) 467-1012
hammond@cs.northwestern.edu
ndnichols@cs.northwestern.edu

Abstract

In this paper we describe an automated system, and its attendant set of techniques and tools, that is able to generate novel multimedia experiences. Using existing online sources, external textual and multimedia repositories, and user preferences, the system builds a customized audio/visual experience for the user. We discuss one application in detail: News at Seven, an automatically generated, personalized news show. Beginning with a set of user preferences, the system is able to find relevant text, process that text, and supplement it with images, video, and blogger responses. The final output of the system is an online Flash presentation that uses animated avatars with generated speech and is modeled after traditional nightly news broadcast. We see this work as the beginning of an overall approach to machine-generated content.

1. Introduction

Driven by increases in bandwidth, advances in encoding/decoding technology, and changes in users' expectations, multimedia has exploded on the web. At the same time, technologies broadly grouped under the "Web 2.0" umbrella have made online mashups and remixes common. Some of these mashups incorporate multimedia. For example, the proof-of-concept YouTunes uses Yahoo! Pipes [36] to generate an RSS feed of music videos for the most popular songs on iTunes; these system use standard search and aggregation techniques to collect and assemble human-created multimedia. While YouTunes and similar mashups are certainly interesting, they do not actually generate new multimedia content; their output is typically XML or HTML with links to a few multimedia elements. We believe that systems can now go beyond simply embedding previously existing media in text; we can create systems that are capable of generating original, compelling multimedia experiences.

This machine-generated content can have a number of advantages over human-created content. First, the content has all the usual benefits of multimedia. It can be delivered to a variety of devices with varying screen sizes and capabilities, it can be viewed in a more passive manner, it can incorporate interesting or exciting audio and visuals, and it can present inherently audio/visual content (i.e., a movie trailer or new album.) Second, because the multimedia is being generated it can also be customized; for example, it can take into account a user's preferences and the capabilities of the machine it's being shown on. Finally, the generated multimedia provides a much richer information experience. A piece of video tends to only have a few pieces of metadata associated with it; for instance, it may have a title, a few keywords, and a sentence-long description. In contrast, a piece of machine-generated multimedia has information about every one of its component pieces; this can include how the content was created, the script, information about the incorporated media and blogger responses, when and for whom the content was created, etc. This supplementary information allows for much richer searching and indexing, a notoriously difficult problem when working with multimedia [6].

News at Seven is our first attempt at building a system that meets both these challenges. It takes an explicit set of user-preferences, and finds news content that might be interesting to that user. Our notion of news is defined broadly, and currently includes options like traditional news, gadget news, celebrity gossip, movie releases, local weather forecasts and Facebook [11] activity. News at Seven is able to draw in news from a wide variety of textual sources, and parse it into homogenous internal data structures; it can then enrich the stories with additional videos, images, and blog entries. The system can also formulate these structures in different ways, so for example, a piece about a new movie release is presented very differently than a story about the war in Iraq. Finally, the system can present the entire news show for a user, using animated avatars, generated speech, and the supplementary media. This final audio/visual presentation, which feels something like a traditional televised news broadcast, is generated entirely for the user.

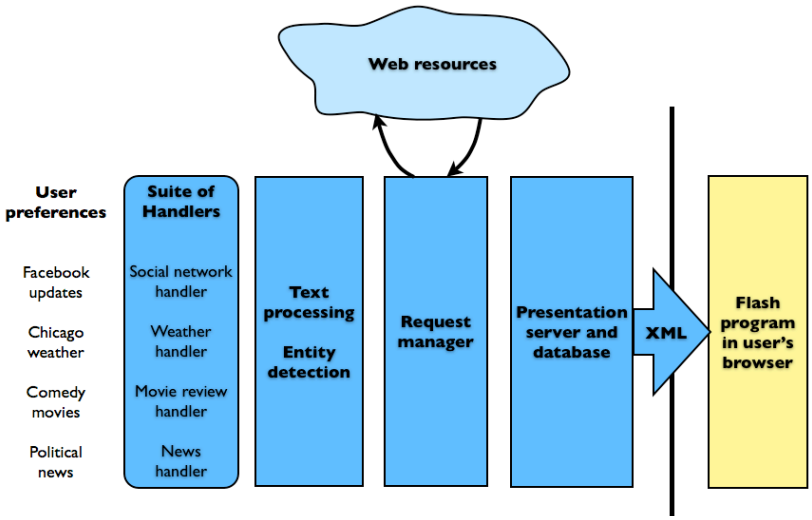
2. Prior Work

There have been a few previous attempts at delivering news using virtual anchors. The most well-known system is probably Ananova [1][3], an avatar from the shoulders up who read news stories using a computer generated voice.

978-0-7695-3529-6/09 $25.00 © 2009 IEEE
DOI 10.1109/ACHI.2009.66

IEEE computer society

Authorized licensed use limited to: Northwestern University. Downloaded on May 7, 2009 at 11:46 from IEEE Xplore. Restrictions apply.



The stories were chosen and annotated by a human producer to tell Ananova which facial expressions to make while reading the story, and the only way for the user to express preferences was by choosing amongst four news channels. While the site is still active, the company was purchased and the focus now is on presenting text news to web browsers and cell phones instead of using an avatar.

Sprint followed Ananova's lead by creating Chase Walker [7]. Chase Walker made the virtual newscast interactive; their model allowed the user to interrupt the anchor, skipping to a different story or moving deeper into the current story. Chase Walker was a more active experience than Ananova, and users apparently even had some voice control over the news show. Unfortunately, Chase Walker also seems to now be defunct.

Perhaps the most similar projects are the related Web2TV [22] and Web2Talkshow [21] systems. Like News at Seven, these systems start with an online textual news story and eventually produce an avatar-hosted news show. Unlike News at Seven, however, these systems do not discover additional related media; instead, they rely strictly on the media already present in a story, which is often only a few static images. Web2Talkshow was also able to do some dialogue creation, but the generated dialogue gives the strong impression of simply filling in joke templates, and generating this dialogue in the first place requires some by-hand story processing.

On the news aggregation side, many news and blog sites provide sources for users to search and browse news. Sites like Google News [15] suggest particular stories for users, and DailyMe [10] provides a personalized newspaper service. My Yahoo![34] allows users to change the background, layout, and content of the site. Other sites, like Technorati [32], allow users to subscribe to an RSS feed for newly posted blogs on topics they are interested in.

There are also systems that read RSS feeds using text-to-speech. Odiogo [27], for example, allows users to subscribe to podcasts that were automatically generated from existing textual RSS feeds. Other systems, like NewsAloud [25], run as a client-side Windows application and generate the speech on the user' machine. Neither of these systems have any sort of preference model (beyond choosing amongst RSS feeds) and neither appears to go beyond simply running a text-to-speech engine over the story.

While individual pieces of the system--like automated avatar-delivered presentations, customized online news sources, and generated speech for RSS feeds--have existed previously, News at Seven is the first application to combine these disparate components into a single system. Furthermore, we have designed News at Seven to avoid many of the weaknesses of these previous efforts while also offering new capabilities.

We believe that while previous systems may have prefigured some of the core functionality of News at Seven, our system is the first that is genuinely able to create novel and compelling multimedia news content. Structurally, News at Seven is split into content generation and show presentation modules (see figure above).

3. Content Generation

The content generation module is primarily composed of a collection of handlers, each capable of generating and producing a different type of segment. For example, there is one handler that produces movie reviews, another that builds traditional news stories, and a third that creates celebrity gossip segments. Before going into detail about the differences between these handlers, we will first discuss the subsystems that the handlers have in common. These include the text processing, entity detection, media discovery and selection, and opinion finding systems.

3.1 Text processing

The first capability of the text-processing subsystem is shortening stories; online news stories are often 1000 words or more in length, but broadcast news shows present stories that may only be a minute or two long. It's important, however, that the system doesn't cut the story off in the

Authorized licensed use limited to: Northwestern University. Downloaded on May 7, 2009 at 11:46 from IEEE Xplore. Restrictions apply.

middle of an idea. Our current algorithm for shortening the text is a straight-forward one. Starting at the beginning of the story, the system adds successive paragraphs until it has at least 500 characters. This is obviously a basic technique, but it works well due to the fact that writers know where their own natural idea breaks are, and punctuate with paragraphs accordingly.

Often online news stories will have sentences like " 'Since it was a really cheap karaoke machine, a bunch of the words were spelled wrong,' Ke said." When a person reads this text, they see the initial quote-marks and understand they are reading a direct quote. When listening to generated speech read the same content, however, listeners may not understand they are hearing a quote and become confused about who is saying what. Introducing quotations appropriately is a known requirement in writing text for news anchors [2][4] and this need is compounded by the flat-affect text-to-speech voices we use. To help combat this potential confusion, the system is able to automatically rearrange many of these phrasings; for example, the previous statement would become "Ke said, 'Since it was a really cheap karaoke machine, a bunch of the words were spelled wrong.' "

The voices we use to eventually generate the speech are from Loquendo [18] and NeoSpeech [24], both running on the Microsoft Speech API [20]. Unfortunately, these two sets of voices have different sets of supported abbreviations. For example, the NeoSpeech voices pronounce "Sun., Apr. 28" as "Sunday, April 28th" while the Loquendo voices read it exactly as written. To remove these discrepancies between the voices, the system expands abbreviations itself using a hand-built and regularly updated set of regular expressions and string-replace pairs.

3.2 Entity Detection

To find supplemental media and opinions the system relies on the named entities present in the original text. This requires a robust entity detection system that can quickly locate and recognize entities like people, cities, countries, movies, albums, etc. mentioned in the text.

The named entity detector we built, the Wikipedia Entity Detector (WPED), relies on a list of entities culled from Wikipedia [33]. Using Wikipedia for named entity detection is a natural idea, and other researchers have built similar systems with advanced disambiguation strategies [5][9]. Our system uses a simpler approach that fits well with News at Seven's requirements.

The WPED begins by processing a configuration file that places a small subset of Wikipedia's categories into a basic semantic hierarchy. For example, the configuration file specifies that all Wikipedia entries in the "Golden Glove Award winners" or "National League All-Stars" categories are baseball players, all baseball players are athletes, and all athletes are people. We use only a subset of Wikipedia and convert Wikipedia's existing categories into our own classification scheme.

Once the WPED has parsed the XML configuration file and knows which Wikipedia categories to include in its database, it loads all of the entries that are a member of any of the tracked categories. The system maintains only a small amount of information for each entry. This information includes the name of the entry, the Wikipedia categories it is a member of, different possible names for that entry, and the number of times it was linked to by other Wikipedia entries. These last two pieces of data are used to resolve two kinds of confusion: multiple names can all refer to one entry, and one name can refer to multiple entries.

The WPED also works very quickly. The system begins with the downloadable version of Wikipedia, parses the fifteen gigabyte XML file, and stores all the information about the entries, categories, references, etc. into a MySQL database. This process takes almost a full day, but is totally automated and only needs to be done every few months. The WPED currently recognizes almost three hundred thousand entities (out of roughly seven million distinct Wikipedia entries) and runs as a web service that returns its results as XML.

3.3 Supplemental Media

Similar to broadcast television news shows, related videos (from YouTube [37]) and images from (Google Image Search [14]) are used to bring a News at Seven story to life. Finding good b-roll media to play while the anchor reads the story is a challenging problem. Not only does the video or image have to be relevant to the text of the story, it also needs to make sense when played behind the anchor. For instance, if the story is about a roadside car bomb attack in Iraq, a video of Presidential candidates discussing the war and its merits might be interesting or relevant; it would be confusing, however, for viewers to see a muted video of candidates talking while the story was being presented.

For a typical news story, the WPED will find ten or so entities. Of course, many of these entities may have only occurred once or twice in a long story, or were only mentioned in passing. Currently, the system discards all the entities that occurred half or less as much as the most frequently occurring entity. If Barack Obama was mentioned six times, Hillary Clinton was mentioned five times, and Washington, DC was mentioned once, the system would discard the Washington, DC entity.

YouTube's search engine is designed to return interesting or entertaining videos related to a user's query. If a user searches YouTube for "Hillary Clinton," she may be interested in video of Clinton speaking or debating, but she is also probably interested in videos about Clinton: parody music videos, political discussion, satirical skits, etc. However, these kinds of videos are confusing when shown in a News at Seven presentation, so we have developed a

Authorized licensed use limited to: Northwestern University. Downloaded on May 7, 2009 at 11:46 from IEEE Xplore. Restrictions apply.

suite of simple filters to make sure the videos the system chooses are of Hillary Clinton, not just about her. For example, the system discards any video in the Comedy category or with keywords like "parody," "spoof", or "Letterman." We also remove videos from the News & Politics category, because these tend to be videos of pundits discussing the subject. We also rely on YouTube's viewers to help eliminate videos of bad quality, and filter videos that are rated lower than three stars or have less than a thousand total views. Finally, a surprising number of videos are spam, and their description will be a long list of popular words like "britney", "sexy", or "porn"; our system throws out any videos with more than five hundred words in their descriptions.

3.4 Alternative Points of View

If a user is interested in a story, there are probably also interested in commentary around and responses to that story. Often, those commentaries and responses--be they sober political analysis, heartfelt replies, or witty rejoinders--are found in the blogosphere. To both enliven the News at Seven presentations and expose the viewer to new perspectives, News at Seven has the ability to automatically find these responses and opinions.

The blog-finding engine of News at Seven uses code originally developed for another project in the lab, Buzz [28]. Buzz is a system that automatically pulls interesting stories from the blogosphere. Beginning with a seed set of topics and story-indicating phrases, it uses blog search engines like Google Blog Search [13] and Technorati to pull back a large number of candidate stories. These candidate stories are then passed through a series of filters, including an emotional affect filter trained on a large corpus of movie and product reviews [29]. After the process is complete, the user is left with a number of compelling stories about the original topics.

3.5 Different types of segments

All of the different handlers share these previous capabilities--processing of the original text, entity detection, supplemental media discovery, and blog opinion finding--in common. However, different handlers eventually produce very different kinds of segments. Some of the differences between segment types are strictly cosmetic, like having a trendy, young anchor talk about new music while an older anchor presents traditional news, or having a Hollywood-themed set for the movie reviews. There is also a set of deeper differences that govern how the segments are scripted and how the supplemental media and blog responses are found. These differences reflect the fact that News at Seven can present a wide variety of information and that different types of information are best presented in different ways. Some of these presentation styles have analogues in broadcast media; for example, our traditional news and movie review segments are based on the televised news and movie review shows that viewers are familiar with. Other types of segments, like celebrity shout-out, have roots in broadcast media but have been modified to work within the limitations and possibilities of machine-generated content. There is a final class of segment type, like updates from friends on Facebook, that have no counterpoint in traditional broadcast media; for these kinds of segments, we are designing wholly new presentation dynamics that viewers find compelling. We will now discuss in more detail the different types of segments, and how they are generated and presented.

Traditional news

Basic news is the default type of segment. It generally consists of a single video or small collection of images, the modified text of the story, and possibly a blog response. These news segments are stories like political or world news, product announcements, sports recaps, etc. These segments are the easiest to generate, and are the simplest to present (more on the system's presentations later.)

Movie reviews

For the movie review segments, we wanted an Ebert and Roeper-type dynamic, with two anchors debating the pros and cons of a new film. Because dialogues like this don't already exist on the internet, the movie review handler has to build the conversation itself.

The conversations built for the movie reviews are

Figure 1: Kaitlin and Sam discussing the new movie *Jumper*

surprisingly convincing. They have occasional grammatical disfluencies where text from a review is inserted into the template, and the two reviewers talk past each other and do not actually address the other's concerns. However, even human reviewers frequently talk past one another, and interstitial material like one reviewer saying "I couldn't disagree with you more" to the other makes the conversation feel very real.

Celebrity shout-out

There is a third type of segment style, used for softer, celebrity gossip pieces, that has information needs similar to the traditional news segment. There is one original

Authorized licensed use limited to: Northwestern University. Downloaded on May 7, 2009 at 11:46 from IEEE Xplore. Restrictions apply.

story, and one associated video. There is a difference in the blogger response, however; rather than having a second anchor respond to the entire story with a single blog posting, this other anchor is constantly interrupting the main anchor with a short opinion whenever a celebrity's name is mentioned.

Like the movie review segments, two avatars talking back and forth helps to keep these celebrity gossip segments lively, and the non sequiturs from the interrupter character are often funny.

Weather

The system can also generate local weather reports for a viewer. The system pulls its local weather from the National Weather Services API [23]. Similar to the movie review segments, the system has a basic set of dialogue templates it uses to convert the raw data returned from the NWS API into text. In our first attempt at building the weather segment handler, these templates would convert the data directly, and produce sentences like "Today the high will be 45, the low 28. The chance of rain is 82%."

Facebook

The Facebook segments are one of the most exciting features of News at Seven because they are indicative of the hyper-personalization that we want the system to be able to perform.

Facebook already provides its users with a news feed, but this feed has a few problems that keep it from being useful for News at Seven. The feed is intended to tell users when their friends' profiles have changed; the problem is that the feed quickly gets filled with minutiae of friends adding different friends, commenting on other friends' pictures, adding applications, etc. This "news" is often not even interesting to skim through, and it wouldn't be any more interesting to hear read to you. Furthermore, the news feed is not currently exposed through Facebook's public API, which we wanted to use because it allows Facebook itself to manage all the potential privacy issues.

After a user adds the News at Seven Facebook application, our system begins taking daily snapshots of all their friends' profiles. Specifically, our system stores each friend's favorite movies, books, music, and quotes, and a list of the photo albums each friend has uploaded.

3.7 Final preparation

At this point, every segment has been scripted by the text-processing and blog-finding systems, and has associated pieces of media. Although different types of segments are created in different ways (taking text from an existing story vs. scripting a movie review dialogue, for example), they all are eventually stored in the same internal representations.

The last step the system needs to do is convert these internal representations into something that can be actually presented to the user in Adobe Flash. The most important part of this is generating the speech, but there is also work to be done on indexing the segment to be retrieved later.

Once the segment has replaced its textual script of what the anchors will say with a representation of the recorded speech and its lip-syncing information, the segment is essentially complete and assigned a unique ID. The only thing left to do for the segment is to index and store it so that it can be retrieved quickly when the user requests it.

To find this kind of information, the system has a second level of database stored in a MySQL installation. The database stores segment categories (like entertainment or world news), segment types, creation dates, etc. along with the associated segment ID. If the system needs to retrieve an entertainment story for a user the system is able to quickly find a list of possible segment IDs in that category, pick the newest one, and then pull that off the disk.

4. Show Presentation

An XML file is the interface between the server-side component, written in Python, and the Flash player that presents the actual show. This XML represents the entire show; for each segment, it has information like the URLs of the supplemental media and sound files, and which anchors and backgrounds to use.

The Flash parses the XML into internal data structures, and begins loading the external assets. Currently, all of the anchors and the backgrounds are stored as separate, external Flash files. This was done to allow easier development of new anchors and sets in the future, and to reduce unnecessary loading (if the show you're watching only has traditional news segments, there is no need for you to download the movie review set.)

5. Future Work

The future plans for News at Seven include development in four main areas: allowing for more personalization and additional sources, making the anchors more expressive physically and verbally, porting the system to other platforms, and allowing for human editors.

5.1 Personalized news

Currently, visitors to our site can choose amongst various news options to build their shows. If the visitor says they like Entertainment news then they will get entertainment stories from Yahoo! News, but there is currently no way for them to supply their own source for entertainment news. There are two main issues with supporting arbitrary RSS feeds.

The first is that if we want the system to read more of the story than the small snippet that's included in the RSS feed, the system needs to be able to follow the link in the feed and pull out the actual content from the page. The second problem is simply one of scale. Although a story only takes roughly twenty seconds to have its text processed, supplemental media found, and speech generated this isn't fast enough to support thousands of RSS feeds.

Authorized licensed use limited to: Northwestern University. Downloaded on May 7, 2009 at 11:46 from IEEE Xplore. Restrictions apply.

5.2 Anchor expressiveness

The voices we are currently using for our anchors do an adequate job of making the anchors understandable, but they tend to drone on and are generally flat sounding. The voices technically support markup, like changing the pitch or rate of the speech, but in practice these controls are not nearly fine-grained enough and are sometimes ignored by the voice engines entirely. We are evaluating other voice packages in the hopes of finding more expressive speech.

The anchors are also not very physically expressive. They have a set of animations they can perform (like shrugging their shoulders or pointing to the screen) while they speak, and we have performed some preliminary research into learning when the anchors should gesture [26].

6. Conclusion

We believe that we can now create systems that produce compelling machine-generated multimedia content. Increases in bandwidth and computational power, more accessible online multimedia repositories, and a pervasive Web 2.0 "mix and mash" ethos have all combined to make multimedia content specifically generated for just one user a very real and exciting possibility. The machine-generated multimedia possesses all the benefits of usual multimedia, while also being cheaper and faster to create, easier to index, and hyper-personalizable. Systems that generate multimedia content need to be facile in both generating wholly new content and incorporating existing content, fast and reliable, and above all, produce interesting and compelling pieces of multimedia.

While News at Seven is our first attempt at such a system, it will not be our last. We intend on using both actual subsystems from News at Seven--like the entity detector and supplemental media finder--and techniques that have worked well to help power new systems. We are excited about the future of this technology, and we look forward to helping craft the future of machine-generated content.

7. Acknowledgments

We are grateful to the National Science Foundation for sponsoring this work under grant number 0535231.

8. References

[1] Ananova, http://www.ananova.com, 2008.

[3] Blank, Douglas. AI Update. Intelligence, Volume 12, Issue 1 (Spring 2001), ACM Press, 6-12.

[4] Block, Mervin and Durso, Joe Jr. "Writing News for TV & Radio". Bonus Books, Chicago, IL, 1998.

[5] Bunescu, R., Pasca, M. "Using Encyclopedic Knowledge for Named Entity Disambiguation". 11th Conference of the European Chapter of the Association for Computational Linguistics, 2006.

[6] Chang, S., Huang, Q., Huang, T., Puri, A., Shahraray, B. Multimedia Search and Retrieval. In Advances in Multimedia: Systems, Standards, and Networks, A. Puri and T. Chen (eds.). New York: Marcel Dekker, 1999.

[7] Chase Walker (Sprint and Headpedal), http://www.sprint.com/ar/chase, 2006.

[8] Clearforest Text Analytics Solutions. http://www.clearforest.com/, 2008.

[9] Cucerzan, S. "Large-Scale Named Entity Disambiguation Based on Wikipedia Data". Joint Conference on Empirical Methods in NLP and Computational Natural Language Learning, 2008

[10] DailyMe. http://www.dailyme.com/, 2008.

[11] Facebook, http://www.facebook.com, 2008.

[12] Flickr, http://www.flickr.com, 2008.

[13] Google Blog Search, http://blogsearch.google.com, 2008.

[14] Google Images, http://images.google.com, 2008.

[15] Google News, http://news.google.com, 2008.

[16] The Internet Movie Database, www.imdb.com, 2008.

[17] Last.FM, http://www.last.fm, 2008.

[18] Loquendo, http://www.loquendo.com, 2008.

[19] Metacritic, http://www.metacritic.com, 2008.

[21] Nadamoto, A., Hayashi, M., Tanaka, K. "Web2TalkShow: Transforming Web content into TV-program-like Content Based on the Creation of Dialogue". WWW, 2005.

[22] Nadamoto, A. Tanaka, K. "Complementing Your TV-Viewing by Web Content Automatically-Transformed into TV-program-type Content". ACM MM, 2005.

[23] National Weather Service API, http://www.nws.noaa.gov/forecasts/xml/, 2008.

[25] NewsAloud, http://www.nextup.com/NewsAloud/, 2008.

[26] Nichols, N., Liu, J., Pardo, B., Hammond, K., Birnbaum, L. Learning to Gesture: Applying animations To Spoken Text. ACM MM, 2007.

[27] Odiogo, http://www.odiogo.com, 2008.

[28] Owsley, S., Hammond, K., Shamma, D., and Sood S. "Buzz: Telling Compelling Stories". ACM MM Interactive Art, 2006.

[29] Owsley S., Sood S., Hammond K., "Domain Specific Affective Classification of Documents." AAAI Spring Symposium on Computational Approaches to Analysing Weblogs, 2006.

[30] Rotten Tomatoes, http://www.rottentomatoes.com/, 2008.

[31] Salton, G., and Buckley, C. Term-weighting approaches in automatic text retrieval. Information Processing and Management 24(5):513-523, 1988.

Authorized licensed use limited to: Northwestern University. Downloaded on May 7, 2009 at 11:46 from IEEE Xplore. Restrictions apply.