VoiSTV: Voice-Enabled Social TV

Bernard Renger
AT&T Labs - Research
Florham Park, NJ, USA
renger@research.att.com

Junlan Feng
AT&T Labs - Research
Florham Park, NJ, USA
junlan@research.att.com

Ovidiu Dan
Lehigh University
Bethlehem, PA, USA
ovd209@lehigh.edu

Harry Chang
AT&T Labs - Research
Austin, TX, USA
harry_chang@research.att.com

Luciano Barbosa
AT&T Labs - Research
Florham Park, NJ, USA
lbarbosa@research.att.com

ABSTRACT

Until recently, the TV viewing experience has not been a very social activity compared to activities on the World Wide Web. In this work, we will present a Voice-enabled Social TV system (VoiSTV) which allows users to interact, follow and monitor the online social media messages related to a TV show while watching it. Users can create, send, and reply to messages using spoken language. VoiSTV also provides metadata information about TV shows such as trends, hot topics, popularity as well as aggregated sentiment of show-related messages, all of which are valuable for TV program search and recommendation.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces

General Terms

Design, Human Factors.

Keywords

Social TV, social data mining, speech interface, Twitter, IPTV.

1. INTRODUCTION

We have seen the great success in the past three years of online social networks such as Facebook, Twitter, MySpace and LinkedIn. According to a Nielsen report[1], social networks or blog sites are visited by three quarters of global consumers who go online. The phenomenal rise of social networks is also changing the way viewers perceive their TV viewing experience. Until recently, the TV viewing experience mainly centered on friends and family as opposed to the social media crowd on social media sites. However, with the burgeoning demand for Internet-connected TV sets, this lack of social experience during TV viewing is starting to change. Cable providers, software companies, and hardware manufacturers are launching products with social TV functionalities, such as Verizon FiOS, Apple TV, Yahoo! TV, Google TV, Boxee, etc. Most of these systems allow users to read social media messages on the TV. However, the supported social functionalities are in their infancy. For instance, Google TV provides some integration with Twitter but does not show tweets related to what the user is watching. Verizon FIOS, on the other hand, provides show-related tweets but does not appear to use any type of machine learning or data mining techniques to retrieve the most relevant tweets. Moreover, the whole user experience is performed by typing on the remote control, which might be cumbersome for composing longer tweets.

In this work, we will present a Voice-enabled social TV system (VoiSTV) which allows users to interact, follow and monitor the messages (tweets) on Twitter[2] related to a TV show while watching it. More specifically, VoiSTV first collects relevant tweets for each show, by employing a novel bootstrapping algorithm based on machine learning in order to suggest query expansions and determine if a candidate tweet is truly relevant to a given TV show. Second, VoiSTV archives and mines the collected tweets in order to capture trending topics, detect sentiment on the tweets, as well as estimate the popularity of shows. The collected corpus along with the mined metadata is valuable for a number of TV applications such as TV program navigation, search, and recommendation. Third, VoiSTV integrates the archived and mined data into a graphical interface running on the TV Set Top Box (STB) to enable Twitter access capabilities through the TV as well as to provide access to the archived corpus and metadata. From this interface, users can also post tweets using speech and send the recognized speech as a status update.

The remainder of this paper is organized as follows. In Section 2, we present the system architecture and describe the technology components and evaluations. Section 3 demonstrates how the system works using a few scenarios. We summarize the paper in Section 4.

2. VOISTV SYSTEM

Figure 1 shows the architecture of the VoiSTV system. VoiSTV has three major function blocks: Data Manager, Data Mining Module and Application Manager. The Data Manager retrieves tweets relevant to TV shows and archives

[1]http://blog.nielsen.com/nielsenwire/online_mobile/social-media-accounts-for-22-percent-of-time-online/

[2]Twitter is a micro-blogging social networking Web site that has a large and rapidly growing user base [5].

WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0637-9/11/03.

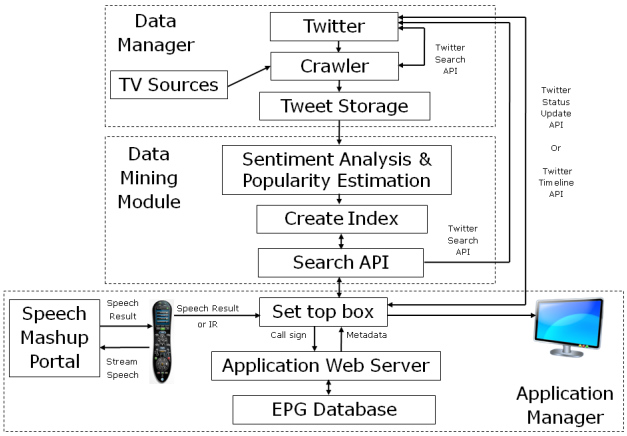

Figure 1: The Architecture of the VoiSTV system.

them. The Data Mining Module indexes the collected tweets, analyzes them, and supports various types of searches. The Application Manager hosts the TV application page which integrates the archived Twitter data, Twitter stream data, and the generated data from the Data Mining Module into an interactive interface on the TV. Below, we elaborate on the key components.

2.1 Data Manager

The Data Manager takes the TV program sources as input and retrieves relevant tweets for each TV show through Twitter's API by issuing queries. The core task of the Data Manager is to collect tweets relevant to each TV show. The main challenge is to obtain relevant tweets with high precision and recall.

Input. The messages on Twitter are in the form of tweets, which are short status updates (of 140 characters or less). While the writing style and the lexicon of tweets are widely varied, much of them are similar to Short Message Service (SMS) text messages. Tweets are often highly ungrammatical, and filled with spelling errors. The 140 character limit also introduces shorthand notations and shortened URLs. There are a few special symbols allowed in tweets: hashtag (e.g., "#obama", a topic tag provided by the user), username (e.g., @twUser), shortened URLs (e.g., "http://bit.ly/9K4n9p"), and the retweet symbol "RT". There are over 50 million tweets per day.

Challenges. There are a number of challenges in obtaining tweets relevant to a given TV show with high precision and recall. First, searching for tweets using only the show title as the keyword phrase might find many tweets not relevant to the TV show. For instance, some shows have titles with generic terms such as "house" and "now". Using such keywords to search for relevant tweets will lead to a low precision collection. In fact, we found that a significant number of shows fall into this category. Second, there is an expression gap between languages used in tweets and authorized resources such as TV program databases. Users on Twitter refer to a TV show in various ways such as using a show-related hashtag or a nickname of the title. Hence, we also face the challenge of low recall. The problem of low recall is more severe for shows with long titles, which most users on Twitter do not use because of the 140 character limit. The third challenge is the shortness and informality of tweets, which tends to make state-of-art text classification techniques fail because of the lack of information. This shortness of tweets coupled with spelling errors and non-grammatical writing styles make finding relevant tweets even more difficult.

Crawler. As shown in Figure 1, the goal of the crawler is to collect relevant tweets. We propose a bootstrapping approach using machine learning algorithms. It starts with a small set of annotated data, where for a given show and a candidate message, we annotate the pair to be relevant or irrelevant. From this annotated data set, we train an initial classifier. The features are designed to capture the association between the TV program and the tweets. Second, using our initial classifier and a large dataset of unlabeled messages, we derive a broader set of features for a second classifier in order to improve the precision. We use nine features for the initial classifier, which capture (among other information) if a tweet contains TV related terms such as "episode", if the first word in the show title mentioned in the tweet is capitalized, if actors and characters from the show are mentioned, as well as the text similarity between the tweet and the Wikipedia page of the corresponding show. The second classifier extends the first one with five new features derived by running the first classifier on a large corpus of unlabeled messages. Details of these features will be published in a future technical paper. The second classifier can also be used to increase the overall recall of the system. Running it on a large dataset can help determine which keywords, hashtags, and user accounts are most commonly mentioned for each show. These strings can then be added to the list of queries we send to Twitter to get more potentially relevant messages. For instance, by running the second classifier on our dataset, we determined that users use the hashtag "#himym" to refer to the show "How I met your mother". We can now use the hashtag to retrieve more messages about the show.

Evaluation. We annotated 3,000 tweets, which were randomly chosen from a tweet collection obtained using 3 show titles (Fringe, Monk, Heroes) as queries. After removing the tweets which our labelers marked as ambiguous, we are left with 2,629 tweets annotated with "Yes" if the message is about the show, or "No" otherwise. We trained the classifiers on tweets associated with two shows and tested on the remaining one. We averaged the precisions over the three possible combinations. For class "Yes" we achieved 80.1% precision for the first classifier and 84.3% for the second classifier.

2.2 Data Mining Module

The Data Mining Module indexes the collected tweets, analyzes them, and supports various types of searches. It provides aggregated statistics such as sentiment analysis, TV show popularity, word cloud, trending topics, and popular messages for each show. Below, we elaborate on the key components.

Index and Search. We have used the SOLR [6] full-text search server to index the tweets. Along with standard TV show information such as author, date, and text, we also store information on sentiment analysis, the ID of the shows mentioned in the message, and a normalized version of the

text used for determining popular messages, as explained below. Our Search API can then query SOLR and return information to the STB. For instance, it returns relevant tweets for a given TV show, ranks the messages based on recency, provides the number of messages about a particular show for a given time window, returns tweets with positive and negative sentiment, generates a word cloud derived from the archived tweets from a given show, etc.. The size of each word in the word cloud is proportional to how often it occurs in the messages about the show. We also filter out noise from the cloud by removing the words which are part of the title of the show and by using a list of stopwords such as "watching", "episode", etc..

Sentiment Analysis. We classified each tweet into one of three sentiment categories: Neutral, Positive, and Negative based on the approach described in [1]. We use a 2-step sentiment analysis method using noisy training data. We first classify tweets as subjective (polar) and objective (non-polar) and further distinguish polar tweets as positive and negative. We collected the training data from three popular Twitter sentiment detection web sites. Our method achieved 81.9% accuracy for subjectivity detection and 81.3% for polarity detection (positive vs. negative). Although we evaluated our algorithm on single-word titles (which are the most ambiguous), it can also correctly find and classify messages for shows with multi-word titles.

Popular Messages. We developed a preliminary approach for estimating the popularity of Twitter messages. We define the popularity of a message to be the number of times we have seen it in the system (usually multiple retweets of the same message). For each message we generate a normalized version by removing the word "RT", removing mentions to usernames (@user), and changing text to lowercase. We then generate a MD5 hash of this normalized version and store it along with the actual text [3]. Two messages with the same content will correspond to the same hash. SOLR is then able to return popular messages from a certain time window. For a given show, tweets are ranked in descending order based on the frequency of the hash.

2.3 Application Manager

As Figure 1 shows, the Application Manager handles the real-time interaction of the user with the system. It accepts speech input from a voice-enabled remote control which is converted into text using the AT&T Speech Mashup Portal [2]. In this section, we will describe the components inside the Application Manager including the STB, Speech Mashup Portal, and the Application Web Server.

STB. The STB is the center point of the system from the user's point of view. Traditionally, the remote control is used for changing channels and navigating TV menus. In our system, users can navigate the TV menus, speak tweets by pressing a TALK button on the voice-enabled remote, see the speech recognition output on the STB/TV, send tweets, reply to tweets and retweet tweets. The STB gets the speech recognition output by using the Speech Mashup Portal. Every time the user switches channels, the STB fetches the TV show information from the Electronic Program Guide (EPG) database, which contains schedule and show information for the upcoming 14 days, and uses the program name to retrieve tweets relevant to the current show through the Search APIs in the Data Mining Module. The STB displays the TV content (high definition video) and the relevant

Figure 2: Program Tweets.

tweets on the same screen. Instead of tweets relevant to the current program, it can also show the users' personal tweets on the same screen as the TV show. The STB also displays metadata information of TV shows provided by the Data Mining Module.

Speech Mashup Portal. The STB streams the users' speech to the Speech Mashup Portal which provides the speech recognition result. The underlying speech recognition engine is AT&T Watson [4]. The acoustic and language models of Watson were trained using SMS messages. The STB displays the recognition result and allows the user to confirm it before sending the tweet using the Twitter status update API.

Application Web Server. The role of the Application Web Server is to host the application web page that runs on the STB. Essentially all the capabilities described above about the STB are possible because of the Application Web Server. The Application Web Server also runs a web service to handle database requests to the EPG database for a given channel in order to determine the show title currently playing on that channel.

3. DEMONSTRATION

This section will focus on what the user sees on the TV screen and how the user interacts with the TV and Twitter. The main menu options are Program Tweets, My Tweets, Program Trends, General Trends, and Send Tweets. The menu options are shown on the left and navigated by using the up and down arrows. Each menu option except the Send Tweets option has more options which can be reached by hitting the OK button. This reveals another level of menus with more options. The context window is to the right of the menu options. At the top level, the context window is the TV program being watched. As the user uses the arrow keys to move up or down, the bottom bar updates with relevant information. This section will focus on the Program Tweets, Send Tweets, and Program Trends options.

3.1 Option One: Program Tweets

For the Program Tweets option, the tweets for the current program are shown in the bottom bar as shown in Figure 2. Only one tweet is shown at a time. The user can see more tweets by using the right and left arrows. The left most tweet is the most recent tweet and the tweets are numbered so the user can keep track of which tweet is being displayed. As new tweets arrive for the current program being watched,

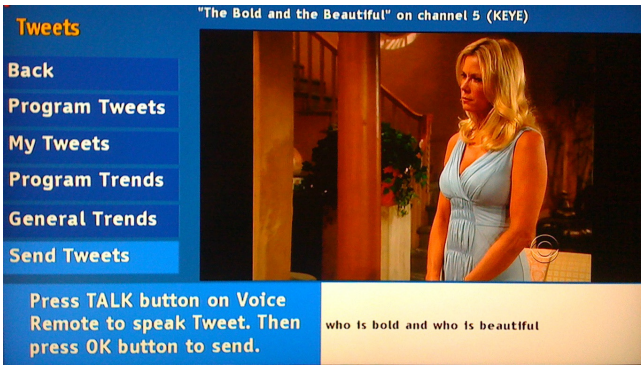

Figure 3: Send Tweet.

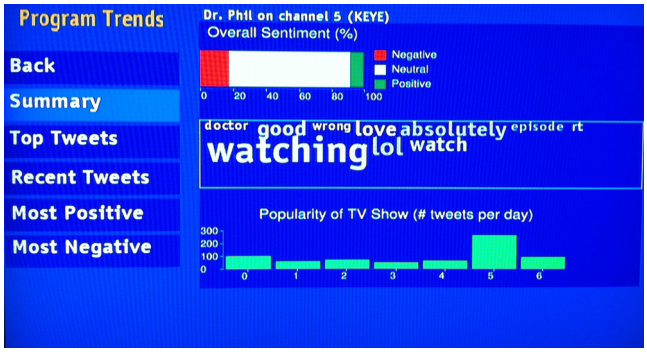

Figure 4: Program Trends Summary.

the numbering will be updated appropriately while keeping the currently viewed tweet in the bottom bar. For example, if the user is viewing tweet 1 of 20 and a new one arrives, the numbering changes to 2 of 21 so that it is clear that a new one arrived and can be viewed by moving to the left using the left arrow key. If the user changes the channel, the video will change to the current program on the selected channel and the tweets in the bottom bar will be replaced with tweets relevant to the new program.

3.2 Option Two: Send Tweets

For the Send Tweets options, the bottom bar shows instructions on how to create a tweet using speech as shown in Figure 3. The instructions indicate that the user should press the TALK button on the remote and speak the message and then press the OK button to send the tweet. The speech result will appear in the bottom bar moments after speaking into the remote so the user has the option of confirming the speech result and can press the OK button to send the tweet if it is acceptable. Otherwise, the user can speak into the remote again to overwrite the last result.

3.3 Option Three: Program Trends

As described above, the data mining backend is used to retrieve the most relevant program tweets. The Program Trends option gives the user access to much more information that is generated by the Data Mining component of our system. For example, the Summary option for Program Trends as shown in Figure 4 provides a chart showing the results of sentiment analysis (negative, neutral, and positive), a word cloud containing words that occur most frequently in tweets about the current program, and a popularity chart that shows the number of tweets in the last week about the current program. The Top Tweets option shows the tweets about the current program that are retweeted the most. Recent tweets are the same tweets that were shown in the top-level Program Tweets option. Finally, the Most Positive and Most Negative are the most positive and negative tweets about the show from the sentiment analysis.

4. SUMMARY

This paper describes VoiSTV, a Voice enabled Social TV system. It gives users access to online social media on the TV while watching TV. Users can input tweets to be sent using voice. We discussed the challenges, described the key components and user interface for interacting with the system. We also evaluated our Data Manager components. Designing a study to evaluate the overall performance of VoiSTV is a challenging task and will be part of our future work.

5. ACKNOWLDGEMENTS

We would like to thank Mazin Gilbert, Behzad Shahraray, and Juergen Schroeter for their support and inspiring discussions on the ideas presented in this paper.

6. REFERENCES

[1] L. Barbosa and J. Feng. Robust Sentiment Detection on Twitter from Biased and Noisy Data. In *Proceedings of Coling*, 2010.

[2] G. Di Fabbrizio, T. Okken, and J. Wilpon. A speech mashup framework for multimodal mobile services. In *Proceedings of the 2009 international conference on Multimodal interfaces*, pages 71–78. ACM, 2009.

[3] H. Dobbertin. The status of MD5 after a recent attack, 1996.

[4] V. Goffin, C. Allauzen, E. Bocchieri, D. Hakkani-Tur, A. Ljolje, S. Parthasarathy, M. Rahim, G. Riccardi, and M. Saraclar. The AT&T Watson speech recognizer. In *Proceedings of ICASSP*, pages 1033–1036, 2005.

[5] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010.

[6] D. Smiley and E. Pugh. Solr 1.4 Enterprise Search Server. 2009.