TwitterFM: An Experimental Application in Entertainment and Social Broadcasting

Theodoros Kalaitzidis,, Nikitas Sgouros

TwitterFM: An experimental application in entertainment and social broadcasting

Theodoros Kalaitzidis, Nikitas M. Sgouros

Department of Digital Systems, University of Piraeus
18534, Piraeus, Greece
sgouros@unipi.gr

Abstract. We describe TwitterFM a web application that explores ways by which social broadcasting via Twitter can be enriched with entertainment features. The system views Twitter as a dynamic set of 'radio stations', each one transmitting under a keyword (hashtag). The user can 'tune in' to the radio stations he wants. Once this happens all messages transmitted by this station are rendered into speech via TTS accompanied with music selected by the user and/or the system. TwitterFM analyzes the affective content of each rendered message and colors accordingly the sub-window in which it is displayed.

Keywords: Entertainment Applications, Microblogging, Social Broadcasting.

1 System Description

Entertainment is inherently a social process as it provides an experiential setting for the elicitation of affective reactions and behaviors. Social broadcasting, on the other hand, has greatly enabled access to information by providing decentralized models for the dissemination of news and opinions and by allowing information diffusion through social processes such as content sharing and retransmission. Combining entertainment and social broadcasting can create novel media forms that facilitate access and consumption of information and provide media-enriched, socially influenced and customized solutions for content presentation. We describe TwitterFM [1] a web application that explores ways by which entertainment processes can enrich social broadcasting and support microblogging environments such as Twitter. Twitter supports a light and informal style of communication that we believe is suitable for casual interaction with the user similar to what is the case with entertainment radio.

TwitterFM views Twitter as a dynamic set of 'radio stations', each one transmitting under a keyword (hashtag). The user can select the hashtags she is interested in and organize them in various radio bands. She can then 'tune in' to the radio station she wants. Consequently all messages transmitted by this station are turned into speech via TTS and rendered accompanied with music selected by the user and/or the system. Furthermore, the system analyzes the affective content of each rendered message and colors accordingly the sub-window in which it is displayed. TwitterFM employs a mashup architecture combining five web APIs: Twitter, Alchemy, TinySong, LastFM

and YouTube. The Twitter API provides the text messages that are used as input to the system. In particular, it provides the keywords that generate significant amount of interest at the current point in time and the original text of the messages that correspond to the keywords selected by the user. The Alchemy API is used for affective analysis of the content of each Twitter message used by the system. TinySong offers a simple interface for collecting links to the music tracks that match with the keywords selected by the user. It is used for identifying the tracks that the user wants to enter in his playlist and to collect additional info on these tracks that will be used for retrieving them from YouTube. LastFM API is used for retrieving similar music and artists to the ones selected by the user in her playlist while YouTube is used to reproduce the tracks selected by the user. In addition, TwitterFM uses its own rudimentary API for coordinating the use of the five APIs described above thereby allowing the user to view/change Twitter trends and/or look for and suggest appropriate music. The system offers appropriate endpoints for other applications to use the results of TwitterFM in these areas. Finally the system uses the FreeTTS API for converting the text of Twitter messages to speech. TwitterFM seeks to recreate the radio experience when listening to Twitter messages. To this effect it uses a radio metaphor for structuring user interaction. In particular, the system uses an analog radio description of the keywords that are trending in Twitter and the ones selected by the user (see Fig. 1). The messages received by the system are organized in 'bands' similar to the way radio stations are organized in radio bands (FM. MW etc). One of these bands called 'Trends" is generated automatically by the system based on the top trending keywords in Twitter at each point in time. The rest of the bands are determined by the user. In particular, the user selects the keywords that will be included in each of these bands and these keywords appear on the radio screen as 'frequencies'. If the user selects any one of these keywords in the band then the system moves the needle on top of the selection and the system starts collecting Twitter messages containing this hashtag. These messages are then converted into speech with FreeTTS and their affective content is analyzed using Alchemy. A tweet is labeled as 'positive' or 'negative' and the label is accompanied by a numerical value indicating the confidence of Alchemy for the label. Depending on the label and its confidence TwitterFM colors each text either red for negative or green for positive along with all the intermediate shades depending on the level confidence. For example in Fig. 1 the middle tweet has been labeled as positive with confidence 0.8 (bright green) compared to the left one which was labeled as positive with confidence 0.2 (light green). The user can create a list of music tracks that wants to be played along with the tweets. The systems plays sequentially the tracks in this list and also allows the user to search for tracks and edit its playlist. TwitterFM constantly monitors the number of remaining tracks in its play list and if this number is less than a threshold it tries to append to the list tracks and artists similar to the ones in the playlist using the LastFM API.

Users found the system original and interesting. The major problem they had involved the actual content of Twitter messages which very often contains web-specific information such as links to web sites that when spoken can be distracting. A solution could involve screening the textual content of each message and removing these links from the spoken part. In addition, due to the limitations of the TTS system, TwitterFM ignores all punctuation marks in the message content. This results in a

spoken interpretation of the message that is flat and monotonous. Both problems expose a more general issue with microblogging systems that convert between different types of media. These systems should allow content creators to prescribe the ways with which their messages will be rendered in different modalities. This could possibly create conflicts with the consumers of these messages. Another issue concerned the placement of spoken messages in the audio stream. Currently, all messages are rendered as soon as they are received and message speech is often mixed with the current audio track. Users suggested rendering the spoken messages between audio tracks similar to what happens to commercials in entertainment radio. In terms of related work there exist a number of apps that read aloud content from social network sites (e.g. iHear, Flipboard or SpokenLayer). However these apps do not enrich the content in some meaningful fashion the way TwitterFM does. TwitterFM is one of a number of web applications that perform post processing on Twitter messages in order to enrich or customize their content. Similar applications with TwitterFM in this category include SocialAudio and SocialRadio. TwitterFM differs from both systems as it provides a radio-band analogy for organizing the hashtags selected by the user and in that it provides a synchronous textual presentation of tweet content enriched with affective analysis. There has been significant interest in the automatic generation of music playlists (e.g. [2]) and, very recently, for original music synthesis in social broadcasting ([3]).

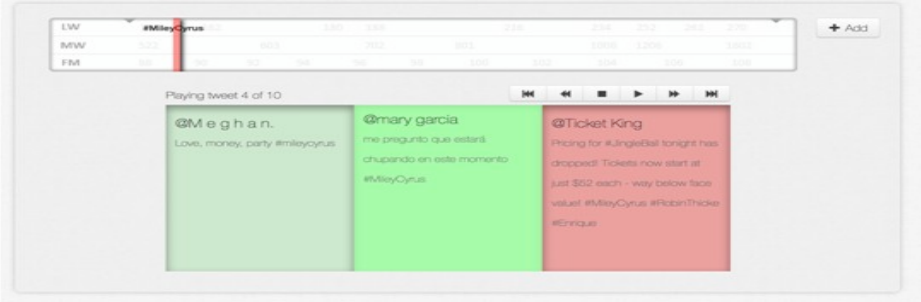

Fig. 1: TwitterFM user interface

Acknowledgements. Presentation of this paper was partially supported by the University of Piraeus Research Center.

References

1. Kalaitzidis, T.: Development of a social TTS application, M.Sc. Thesis, Dept of Digital Systems, University of Piraeus, December (2013)
2. Coelho, F., Devezas, J., Ribeiro, C.: Large-scale Crossmedia Retrieval for Playlist Generation and Song Discovery, Proceedings of the 10th Conference on Open Research Areas in Information Retrieval, Portugal (2013)
3. Moreale, F., de Angeli, A., Miniukovich, A.: TwitterRadio: Translating Tweets into Music, CHI 2014, Canada (2014)