MemeTube: A Sentiment-based Audiovisual System for Analyzing and Displaying Microblog Messages

Cheng-Te Li[1] **Chien-Yuan Wang**[2] **Chien-Lin Tseng**[2] **Shou-De Lin**[1,2]
[1] Graduate Institute of Networking and Multimedia
[2] Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan
{d98944005, sdlin}@csie.ntu.edu.tw {gagedark, moonspirit.wcy}@gmail.com

Abstract

Micro-blogging services provide platforms for users to share their feelings and ideas on the move. In this paper, we present a search-based demonstration system, called MemeTube, to summarize the sentiments of microblog messages in an audiovisual manner. MemeTube provides three main functions: (1) recognizing the sentiments of messages (2) generating music melody automatically based on detected sentiments, and (3) produce an animation of real-time piano playing for audiovisual display. Our MemeTube system can be accessed via: http://mslab.csie.ntu.edu.tw/memetube/ .

1 Introduction

Micro-blogging services such as Twitter[1], Plurk[2], and Jaiku[3], are platforms that allow users to share immediate but short messages with friends. Generally, the micro-blogging services possess some signature properties that differentiate them from conventional weblogs and forum. First, microblogs deal with almost *real-time messaging,* including instant information, expression of feelings, and immediate ideas. It also provides a source of crowd intelligence that can be used to investigate common feelings or potential trends about certain news or concepts. However, this real-time property can lead to the production of an enormous number of messages that recipients must digest. Second, micro-blogging is *time-traceable*. The temporal information is crucial because contextual posts that appear close together are, to some extent, correlated. Third, the style of micro-blogging posts tends to be *conversation-based* with a sequence of responses. This phenomenon indicates that the posts and their responses are highly correlated in many respects. Fourth, micro-blogging is *friendship-influenced*. Posts from a particular user can also be viewed by his/her friends and might have an impact on them (e.g. the empathy effect) implicitly or explicitly. Therefore, posts from friends in the same period may be correlated sentiment-wise as well as content-wise.

We leverage the above properties to develop an automatic and intuitive Web application, MemeTube, to analyze and display the sentiments behind messages in microblogs. Our system can be regarded as a sentiment-driven, music-based summarization framework as well as a novel audiovisual presentation of art. MemeTube is designed as a search-based tool. The system flow is as shown in Figure 1. Given a query (either a keyword or a user id), the system first extracts a series of relevant posts and replies based on keyword matching. Then sentiment analysis is applied to determine the sentiment of the posts. Next a piece of music is composed to reflect the detected sentiments. Finally, the messages and music are fed into the animation generation model, which displays a piano keyboard that plays automatically.

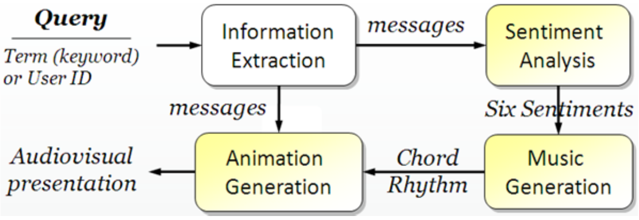

Figure 1: The system flow of our MemeTube.

The contributions of this work can be viewed from three different perspectives.

- From system perspective of view, we demo a novel Web-based system, MemeTube, as a kind of search-based sentiment presentation, musi-

[1] http://www.twitter.com
[2] http://www.plurk.com
[3] http://www.jaiku.com/

Proceedings of the ACL-HLT 2011 System Demonstrations, pages 32–37,
Portland, Oregon, USA, 21 June 2011. ©2011 Association for Computational Linguistics

calization, and visualization tool for microblog messages. It can serve as a real-time sentiment detector or an interactive microblog audio-visual presentation system.

- Technically, we integrate a language-model-based classifier approach with a Markov-transition model to exploit three kinds of information (i.e., contextual, response, and friendship information) for sentiment recognition. We also integrate the sentiment-detection system with a real-time rule-based harmonic music and animation generator to display streams of messages in an audiovisual format.
- Conceptually, our system demonstrates that, instead of simply using textual tags to express sentiments, it is possible to exploit audio (i.e., music) and visual (i.e., animation) cues to present microblog users' feelings and experiences. In this respect, the system can also serve as a Web-based art piece that uses NLP-technologies to concretize and portray sentiments.

2 Related Works

Related works can be divided into two parts: sentiment classification in microblogs, and sentiment-based audiovisual presentation for social media. For the first part, most of related literatures focus on exploiting different classification methods to separate positive and negative sentiments by a variety of textual and linguistics features, as shown in Table 1. Their accuracy ranges from 60%~85% depending on different setups. The major difference between our work and existing approaches is that our model considers three kinds of additional information (i.e., contextual, response and friendship information) for sentiment recognition.

In recent years, a number of studies have investigated integrating emotions and music in certain media applications. For example, Ishizuka and Onisawa (2006) generated variations of theme music to fit the impressions of story scenes represented by textual content or pictures. Kaminskas (2009) aligned music with user-selected points of interests for recommendation. Li and Shan (2007) produced painting slideshows with musical accompaniment. Hua et al. (2004) proposed a Photo2Video system that allows users to specify incident music that expresses their feelings about the photos. To the best of our knowledge, MemeTube is the first attempt to exploit AI techniques to create harmonic audiovisual experiences and interactive emotion-based summarization for microblogs.

Table 1: Summary of related works that detect sentiments in microblogs.

	Features	Methods
Pak and Paroubek 2010	statistic counting of adjectives	Naive Bayes
Chen et al. 2008	POS tags, emoticons	SVM
Lewin and Pribula 2009	smileys, keywords	Maximum Entropy, SVM
Riley 2009	n-grams, smileys, hashtags, replies, URLs, usernames, emoticons	Naive Bayes,
Prasad 2010	n-grams	Naïve Bayes
Go et al. 2009	usernames, sequential patterns of keywords, POS tags, n-grams	Naive Bayes, Maximum Entropy, SVM
Li et al. 2009	several dictionaries about different kinds of keywords	Keyword Matching
Barbosa and Feng 2010	retweets, hashtag, replies, URLs, emoticons, upper cases	SVM
Sun et al. 2010	keyword counting and Chinese dictionaries	Naive Bayes, SVM
Davidov et al. 2010	n-grams, word patterns, punctuation information	*k*-Nearest Neighbor
Bermingham and Smeaton 2010	n-grams and POS tags	Binary Classification

3 Sentiment Analysis of Microblog Posts

First, we develop a classification model as our basic sentiment recognition mechanism. Given a training corpus of posts and responses annotated with sentiment labels, we train an n-gram language model for each sentiment. Then, we use such model to calculate the probability that a post expresses the sentiment s associated with that model:

$$Pr(p|s) = Pr(w_1, \cdots, w_m|s) \prod_{i=1}^{m} Pr\left(w_i \middle| w_{i-(n-1)}, \cdots, w_{i-1}, s\right),$$

where w is the sequence of words in the post. We also use the common Laplace smoothing method.

For each post p and each sentiment $s \in S$, our classifier calculates the probability that such post expresses the sentiment $Pr(s|p)$ using Bayes rule:

$$Pr(s|p) = Pr(s)Pr(p|s)$$

$Pr(s)$ is estimated directly by counting, while $Pr(p|s)$ can be derived by using the learned language models. This allow us to produce a distribution of sentiments for a given post p, denoted as S_p.

However, the major challenge in the microblog sentiment detection task is that the length of each post is limited (i.e., posts on Twitter are limited to 140 characters). Consequently, there might not be enough information for a sentiment detection system to exploit. To solve this problem, we propose to utilize the three types of information mentioned earlier. We discuss each type in detail below.

3.1 Response Factor

We believe the sentiment of a post is highly correlated with (but not necessary similar to) that of responses to the post. For example, an angry post usually triggers angry responses, but a sad post usually solicits supportive responses. We propose to learn the correlation patterns of sentiments from the data and use them to improve the recognition.

To achieve such goal, from the data, we learn the probability $P(Sentiment_{post}|Sentiment_{response})$, which represents the conditional probability of a post given responses. Then we use such probability to construct a transition matrix M_r, where $M_{r_{ij}}$ $=P(Sentiment_{post} = j \mid Sentiment_{response} = i)$.

With M_r, we can generate the adjusted sentiment distribution of the post S'_p as:

$$S'_p = \alpha \times \frac{\sum_{i=1}^{k} W_{r_i} S_{r_i} M_r}{k} + (1-\alpha) S_p ,$$

where S_p denotes the original sentiment distribution of the post, and S_{r_i} is the sentiment distribution of the i^{th} response determined by the abovementioned language model approach. In addition, $W_{r_i} = 1/(t_{response_i} - t_{post})$ represents the weight of the response since it is preferable to assign higher weights to closer responses. There is also a global parameter α that determines how much the system should trust the information derived from the responses to the post. If there is no response to a post, we simply assign $S'_p = S_p$.

3.2 Context Factor

It is assumed that the sentiment of a microblog post is correlated with the author's previous posts (i.e., the 'context' of the post). We also assume that, for each person, there is a sentiment transition matrix M_c that represents how his/her sentiments change over time. The $(i,j)^{th}$ element in M_c represents the conditional probability from the sentiment of the previous post to that of the current post: $P(Sentiment(P_t) = j \mid Sentiment(P_{t-1}) = i)$.

The diagonal elements stand for the consistency of the emotion state of a person. Conceivably, a capricious person's diagnostic $M_{c_{ii}}$ values will be lower than those of a calm person. The matrix M_c can be learned directly from the annotated data.

Let S_t represent the detected sentiment distribution of an existing post at time t. We want to adjust S_t based on the previous posts from $t - \Delta t$ to t, where Δt is a given temporal threshold. The system first extracts a set of posts from the same author posted from time $t - \Delta t$ to t and determines their sentiment distributions $\{S_{t_1}, S_{t_2}, \ldots, S_{t_k}\}$, where $t - \Delta t < t_1, t_2, \ldots, t_k < t$ using the same classifier. Then, the system utilizes the following update equation to obtain an adjusted sentiment distribution S'_t:

$$S'_t = \alpha \times \frac{\sum_{i=1}^{k} W_{t_i} S_{t_i} M_c}{k} + (1-\alpha) S_t ,$$

where $W_{t_i} = 1/(t - t_i)$. The parameters W_{t_i}, k, α are defined similar to the previous case. If there is no post in the defined interval, the system will leave S_t unchanged.

3.3 Friendship Factor

We also assume that the friends' emotions are correlated with each other. This is because friends affect each other, and they are more likely to be in the same circumstances, and thus enjoy/suffer similarly. Our hypothesis is that the sentiment of a post and the sentiments of the author's friends' recent posts might be correlated. Therefore, we can treat the friends' recent posts in the same way as the recent posts of the author, and learn the transition matrix M_f , where $M_{f_{ij}} = P(Sentiment_{user}(P_t) = j \mid Sentiment_{user's\,friend}(P_{t-1}) = i)$, and apply the technique proposed in the previous section to improve the recognition accuracy.

However, it is not necessarily true that all friends have similar emotional patterns. One's sentiment transition matrix M_c might be very different from that of the other, so we need to be careful when using such information to adjust our recognition outcomes. We propose to only consider posts from friends with similar emotional patterns.

To achieve our goal, we first learn every user's contextual sentiment transition matrix M_c from the data. In M_c, each row represents a distribution that sums to one; therefore, we can compare two matrixes M_{c_1} and M_{c_2} by averaging the symmetric KL-divergence of each row. That is,

$$
\begin{aligned}
&Similarity(M_1, M_2) \\
&= Average_{i=1}^{n} KL\big(Row(M_1, i), Row(M_2, i)\big).
\end{aligned}
$$

Two persons are considered as having similar emotion pattern if their contextual sentiment transition matrixes are similar. After a set of similar friends are identified, their recent posts (i.e., from $t - \Delta t$ to t) are treated in the same way as the posts by the author, and we use the method proposed previously to fine-tune the recognition outcomes.

4 Music Generation

For each microblog post retrieved according to the query, we can derive its sentiment distribution (as a vector of probabilities) by using the above method. Next, the system transforms every sentiment distribution into an affective vector comprised of a valence value and an arousal value. The valence value represents the positive-to-negative sentiment, while the arousal value represents the intense-to-silent level.

We exploit the mapping from each type of sentiment to a two-dimensional affective vector based on the two-dimensional emotion model of Russell (1980). Using the model we extract the affective score vectors of the six emotions (see Table 2) used in our experiments. The mapping enables us to transform a sentiment distribution S_p into an affective score vector by weighted sum approach. For example, given a distribution of (Anger=20%, Surprise=20%,Disgust=10%, Fear=10%, Joy=10%, Sadness=30%), the two-dimensional affective vector can be computed as 0.2*(-0.25, 1) + 0.2*(0.5, 0.75) + 0.1*(-0.75, -0.5) + 0.1*(-0.75, 0.5) + 0.1*(1, 0.25) + 0.3*(-1, -0.25). Finally, the affective vector of each post will be summed to represent the sentiment of the given query in terms of the valence and arousal values.

Table 2: Affective score vector for each sentiment label.

Sentiment Label	Affective Score Vector
Anger	(-0.25, 1)
Surprise	(0.5, 0.75)
Disgust	(-0.75, -0.5)
Fear	(-0.75, 0.5)
Joy	(1, 0.25)
Sadness	(-1, -0.25)

Next the system transforms the affective vector into music elements through chord set selection (based on the valence value) and rhythm determination (based on the arousal value). For chord set selection, we design nine basic chord sets as {A, Am, Bm, C, D, Dm, Em, F, G}, where each chord set consists of some basic notes. The chord sets are used to compose twenty chord sequences. Half of the chord sequences are used for weakly positive to strongly positive sentiments and the other half are used for weakly negative to strongly negative sentiments. The valence value is therefore divided into twenty levels, and gradually shifts from strongly positive to strongly negative. The chord sets ensure that the resulting auditory presentation is in harmony (Hewitt 2008). For rhythm determination, we divide the arousal values into five levels to decide the tempo/speed of the music. Higher arousal values generate music with a faster tempo while lower ones lead to slow and easy-listening music.

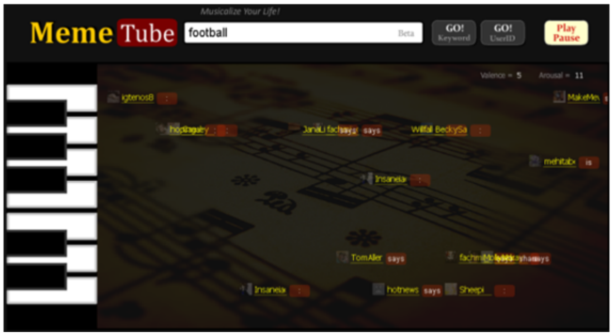

Figure 2: A snapshot of the proposed MemeTube.

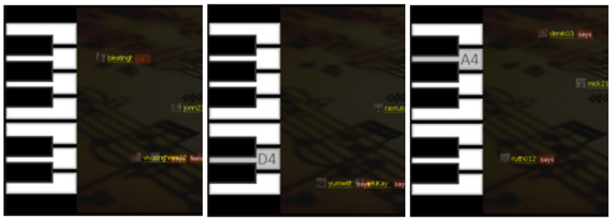

Figure 3: The animation with automatic piano playing.

5 Animation Generation

In this final stage, our system produces real-time animation for visualization. The streams of messages are designed to flow as if they were playing a piece of a piano melody. We associate each message with a note in the generated music. When a post message flows from right to left and touches a piano key, the key itself blinks once and the corresponding tone of the key is produced. The message flow and the chord/rhythm have to be synchronized so that it looks as if the messages themselves are playing the piano. The system also allows users to highlight the body of a message by moving the cursor over the flowing message. A snapshot is shown in Figure 2 and the sequential snapshots of the animation are shown in Figure 3.

6 Evaluations on Sentiment Detection

We collect the posts and responses from every effective user, users with more than 10 messages, of Plurk from January 31st to May 23rd, 2009. In order to create the diversity for the music generation system, we decide to use six different sentiments, as shown in Table 2, rather than using only three sentiment types, positive, negative and neutral, as most of the systems in Table 1 have used. The sentiment of each sentence is labeled automatically using the emoticons. This is similar to what many people have proposed for evaluation (Davidov et al. 2010; Sun et al. 2010; Bifet and Frank 2010; Go et al. 2009; Pak and Paroubek 2010; Chen et al. 2010). We use data from January 31st to April 30th as training set, May 1st to 23rd as testing data. For the purpose of observing the result of using the three factors, we filter the users without friends, the posts without responses, and the posts without previous post in 24 hour in testing data. We also manually label the sentiments on the testing data (totally 1200 posts, 200 posts for each sentiment).

We use three metrics to evaluate our model: accuracy, Root-Mean-Square Error for valence (denoted by RMSE(**V**)) and RMSE for arousal (denoted by RMSE(**A**)). The RMSE values are generated by comparing the affective vector of the predicted sentiment distribution with the affective vector of the answer. Our basic model reaches 33.8% in accuracy, 0.78 in the RMSE(V) and 0.64 in RMSE(A). Note that RMSE≈0.5 means that there is roughly one quarter (25%) error in the valence/arousal values as they range from [-1,1].

Note that the main reason the accuracy is not extremely high is that we are dealing with 6 classes. When we combine angry, disgust, fear, and sadness into one negative sense and the rest as positive senses, our system reaches **78.7%** in accuracy, which is competitive to the state-of-the-art algorithms as shown in the related work section. However, doing such generalization will lose the flexibility of producing more fruitful and diverse pieces of music. Therefore we choose more fine-grained classes for our experiment.

Figure 3 shows the results of exploiting the response, context, and friendship. Note RMSE≈0.5 means that there is roughly one quarter (25%) error in the valence/arousal values as they range from [-1,1]. The results show that considering all three additional factors can achieve the best results and decent improvement over the basic LM model.

Table 3: The results after adding addition info (note that for RMSE, the lower value the better)

	LM	Response	Context	Friend	**Combine**
Accuracy	33.8%	34.7%	34.8%	35.1%	**36.5%**
RMSE(V)	0.784	0.683	0.684	0.703	**0.679**
RMSE(A)	0.640	0.522	0.516	0.538	**0.514**

7 System Demo

We create video clips of five different queries for demonstration, which is downloadable from: http://mslab.csie.ntu.edu.tw/memetube/demo/. This demo page contains the resulting clips of four keyword queries (including *football*, *volcano*, *Monday*, *big bang*) and a user id query *mstcgeek*. Here we briefly describe each case. (1) The video for query term, *football*, was recorded on February 7th 2011, results in a relatively positive and extremely intense atmosphere. It is reasonable because the NFL Super Bowl was played on February 6th, 2011. The valence value is not as high as the arousal value because some fans might not be very happy to see their favorite team losing the game. (2) The query, *volcano*, was also recorded on February 7th 2011. The resulting video expresses negative valence and neutral arousal. After checking the posts, we have learned that it is because the Japanese volcano Mount Asama has continued to erupt. Some users are worried and discussed about the potential follow-up disasters. (3) The query *Monday* was performed on February 6th 2011, which is a Sunday night. The negative valence reflects the “blue Monday” phenomenon, which leads to some heavy, less smooth melody. (4) The term *big bang* turns out to be very positive on both valence and arousal, mainly because, besides its relatively neutral meaning in physics, this term also refers to a famous comic show that some people in Plurk love to watch. We also use one user id as query: the user-id *mstcgeek* is the official account of Microsoft Taiwan. This user often uses cheery texts to share some videos about their products or provide some discounts of their product, which leads to relatively hyped music.

8 Conclusion

Microblog, as a daily journey and social networking service, generally captures the dynamics of the change of feelings over time of the authors and their friends. In MemeTube, the affective vector is generated by aggregating the sentiment distribution of each post; thus, it represents the majority's opinion (or sentiment) about a topic. In this sense, our system can be regarded as providing users with an audiovisual experience to learn collective opinion of a particular topic. It also shows how NLP techniques can be integrated with knowledge about music and visualization to create a piece of interesting network art work. Note that MemeTube can be regarded as a flexible framework as well since each component can be further refined independently. Therefore, our future works are threefold: For sentiment analysis, we will consider more sophisticated ways to improve the baseline accuracy and to aggregate individual posts into a collective consensus. For music generation, we plan to add more instruments and exploit learning approaches to improve the selection of chords. For visualization, we plan to add more interactions between music, sentiments, and users.

Acknowledgements

This work was supported by National Science Council, National Taiwan University and Intel Corporation under Grants NSC99-2911-I-002-001, 99R70600, and 10R80800.

References

Barbosa, L., and Feng, J. 2010. Robust Sentiment Detection on Twitter from Biased and Noisy Data. In Proceedings of International Conference on Computational Linguistics (COLING'10), 36–44.

Bermingham, A., and Smeaton, A. F. 2010. Classifying Sentiment in Microblogs: is Brevity an Advantage? In Proceedings of ACM International Conference on Information and Knowledge Management (CIKM'10), 1183–1186.

Chen, M. Y.; Lin, H. N.; Shih, C. A.; Hsu, Y. C.; Hsu, P. Y.; and Hewitt, M. 2008. Music Theory for Computer Musicians. Delmar.

Hsieh, S. K. 2010. Classifying Mood in Plurks. In Proceedings of Conference on Computational Linguistics and Speech Processing (ROCLING 2010), 172–183.

Davidov, D.; Tsur, O.; and Rappoport, A. 2010. Enhanced Sentiment Learning Using Twitter Hashtags and Smileys. In Proceedings of International Conference on Computational Linguistics (COLING'10), 241–249.

Go, A.; Bhayani, R.; and Huang, L. 2009. Twitter Sentiment Classification using Distant Supervision. *Technical Report*, Stanford University.

Hua, X. S.; Lu, L.; and Zhang, H. J. 2004. Photo2Video - A System for Automatically Converting Photographic Series into Video. In Proceedings of ACM International Conference on Multimedia (MM'04), 708–715.

Ishizuka, K., and Onisawa, T. 2006. Generation of Variations on Theme Music Based on Impressions of Story Scenes. In Proceedings of ACM International Conference on Game Research and Development, 129–136.

Kaminskas, M. 2009. Matching Information Content with Music. In Proceedings of ACM International Conference on Recommendation System (RecSys'09), 405–408.

Lewin, J. S., and Pribula, A. 2009. Extracting Emotion from Twitter. *Technical Report*, Stanford University.

Li, C. T., and Shan, M. K. 2007. Emotion-based Impressionism Slideshow with Automatic Music Accompaniment. In Proceedings of ACM International Conference on Multimedia (MM'07), 839–842.

Li, S.; Zheng, L.; Ren, X.; and Cheng, X. 2009. Emotion Mining Research on Micro-blog. In Proceedings of IEEE Symposium on Web Society, 71–75.

Pak, A., and Paroubek, P. 2010. Twitter Based System: Using Twitter for Disambiguating Sentiment Ambiguous Adjectives. In Proceedings of International Workshop on Semantic Evaluation, (ACL'10), 436–439.

Pak, A., and Paroubek, P. 2010. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In Proceedings of International Conference on Language Resources and Evaluation (LREC'10), 1320–1326.

Prasad, S. 2010. Micro-blogging Sentiment Analysis Using Bayesian Classification Methods. *Technical Report*, Stanford University.

Riley, C. 2009. Emotional Classification of Twitter Messages. *Technical Report*, UC Berkeley.

Russell, J. A. 1980. Circumplex Model of Affect. Journal of Personality and Social Psychology, 39(6):1161–1178.

Strapparava, C., and Valitutti, A. 2004. Wordnet-affect: an Affective extension of wordnet. In Proceedings of International Conference on Language Resources and Evaluation, 1083–1086.

Sun, Y. T.; Chen, C. L.; Liu, C. C.; Liu, C. L.; and Soo, V. W. 2010. Sentiment Classification of Short Chinese Sentences. In Proceedings of Conference on Computational Linguistics and Speech Processing (ROCLING'10), 184–198.

Yang, C.; Lin, K. H. Y.; and Chen, H. H. 2007. Emotion Classification Using Web Blog Corpora. In Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence (WI'07), 275–278.