MusicCommentator: Generating Comments Synchronized with Musical Audio Signals by a Joint Probabilistic Model of Acoustic and Textual Features

Kazuyoshi Yoshii and Masataka Goto

National Institute of Advanced Industrial Science and Technology (AIST)
Central 2, 1-1-1 Umezono, Tsukuba, 305-8568 Ibaraki, Japan
{k.yoshii,m.goto}@aist.go.jp

Abstract. This paper presents a system called *MusicCommentator* that suggests possible comments on appropriate temporal positions in a musical audio clip. In an online video sharing service, many users can provide free-form text comments for temporal events occurring in clips not for entire clips. To emulate the commenting behavior of users, we propose a joint probabilistic model of audio signals and comments. The system trains the model by using existing clips and users' comments given to those clips. Given a new clip and some of its comments, the model is used to estimate what temporal positions could be commented on and what comments could be added to those positions. It then concatenates possible words by taking language constraints into account. Our experimental results showed that using existing comments in a new clip resulted in improved accuracy for generating suitable comments to it.

Keywords: Audio and language processing, user communication modeling, probabilistic music-comment association, comment generation.

1 Introduction

Commenting plays important roles in the entertainment culture of the consumer generated media (CGM). We can access many online content-sharing services such as YouTube (video), MySpace (music), and Flickr (photo) that enable users not only to present their original works but also to comment on works created by others. Users who view the same work can communicate with each other by commenting. For example, users provide tags [1], describe positive or negative reviews [2], and show their agreement or disagreement with specific comments. This kind of user communication has recently been facilitated.

Commenting can be viewed as collaborative creation [3] in an advanced form of user communication: pseudo-synchronized communication. We can see evidence of this in a video sharing service named *Nico Nico Douga* (whose name means "video making people smile" in Japanese).[1] where users can provide comments

[1] Nico Nico Douga reserves over 17 hundred million comments by ten million users. http://www.nicovideo.jp/, http://en.wikipedia.org/wiki/Nico_Nico_Douga.

S. Natkin and J. Dupire (Eds.): ICEC 5709, pp. 85–97, 2009.

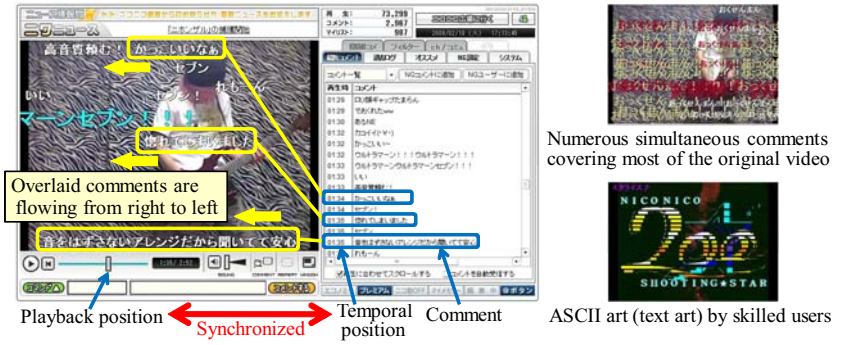

Fig. 1. Screenshots captured from a video sharing service named Nico Nico Douga

at arbitrary temporal positions in the video. A unique feature of this service is that recent comments of many users are overlaid directly onto the video and synchronized to a specific playback time as shown in Fig. 1. This gives users a sense of sharing the viewing experiences. That is, users feel as if they enjoyed the same video together in real time although their comments were provided at different dates and times in the real world. Some kinds of comments therefore can add remarkable and interesting effects to the original video. For instance, we often see *barrages*, where so many identical or similar comments collaboratively made by many users are piled up to a degree that the original video is almost completely hidden, and skilled users create cool drawings[2] by combining many comments (characters) provided at temporally and spatially different positions. Because commenting is much easier than creating new video clips from scratch or reediting existing ones. it can be an important popularized way of creation.

Novice users, however, sometimes feel anxiety when commenting, wondering *Is my comment suitable to the occasion?* Implicit rules seem to be shared among users who collaborate to provide comments, so novice users had better experience what kinds of comments are given by other users and what kinds of temporal events are annotated. Another issue is that video creators can hardly predict what comments will be given to their works, and the fear of being insulted often makes them hesitant about presenting their work to the public, especially the first time they try. Besides these practical issues, from an academic viewpoint, we are interested in the relationships between music and comments and investigate whether it is really possible to generate comments in a human-like fashion.

We therefore developed a system, called *MusicCommentator*, that can generate comments that are most likely to be provided at specific temporal positions in a music video clip. It can help novice users by suggesting comments suitable to the occasion and can help encourage video creators to present their work by letting them virtually experience having comments made about their work. In this study we deal with music, one of the audio parts within video clips, as the first step toward handling all the information in the clips.

[2] ASCII art or text art: http://en.wikipedia.org/wiki/ASCII_art.

The rest of this paper is organized as follows. First, Section 2 introduces related work. Then, Section 3 specifies the commenting problem and Section 4 explains how to build our system. Section 5 reports on our experiments. Finally, Section 6 summarizes the key findings of this paper.

2 Related Work

Several studies have been conducted to predict suitable words to a given musical piece by using the audio signal as input. These studies typically estimate how strongly each word is associated with a given piece. For example, Whitman and Rifkin [4] used a kernel method to predict words that will appear in music reviews. Turnbull *et al.* [5] attempted to associate audio content with semantically meaningful words by using a Gaussian mixture model (GMM) of acoustic features for each word. The output is obtained as sentences by filling slots with predicted words in sentence templates manually prepared beforehand. Bertin-Mahieux *et al.* [6] used an ensemble learning method called AdaBoost to predict social tags, which are free-form text labels at a song or artist level.

Our study differs from the previous studies in two ways. First, we deal with comments that are not given to an entire piece but provided at arbitrary temporal positions in it. It is thus necessary to determine what temporal positions can be annotated in a given piece. Second, we try to generate comments as natural-languages sentences. These goals make our attempt very challenging.

3 Problem Specification

The input data for the MusicCommentator task contains N audio clips (audio tracks of video clips) and their comments provided by users. Note that we focus on audio tracks in this paper even if we deal with video clips. Let n ($1 \leq n \leq N$) be the index of an audio clip. This data is used to train a computational model of commenting. When the system is given a new audio clip (and some existing comments on it), the objective is to add reasonable comments at appropriate temporal positions by using the model. Audio clips are represented as acoustic features and comments are represented as textual features.

1. **Acoustic Features:** We use mel-frequency cepstrum coefficients (MFCCs) and their delta components because these features have been effectively used for characterization of detection of musical genres and moods [7]. Calculating MFCCs at each frame,[3] we can obtain a temporal sequence of feature vectors. Let $\boldsymbol{a}_t^{(n)}$ be a feature vector of frame t in clip n.
2. **Textual Features:** We define three kinds of textual features of comments.
   (a) **Bag-of-Words Features:** These features represent the content of provided comments. We split all Japanese free-form comments into words[4]

[3] A "frame" here is a short duration (256 ms) to be analyzed in an audio clip.

[4] Some words have single morphemes while others have two or more morphemes within them. The inflectional word "loved," for example, consists of the base morpheme "love" and the inflectional morpheme "ed" (past tense).

with a Japanese morphological analyzer called Mecab [8]. Because different words including the same base morpheme are semantically identical, we do not distinguish them. Removing auxiliary words and extracting significant words whose numbers of occurrences are higher than a threshold, we get a vocabulary consisting of V words. Then, we count how many times each word occurs. For example, if a frame contains three comments, "I love it," "It is loved," and "Love song," the average number of occurrences of the verb "love" is 0.66. Let $\boldsymbol{w}_t^{(n)} = \{w_{t,1}^{(n)}, \cdots, w_{t,V}^{(n)}\}$ be a bag-of-words vector, where $w_{t,v}^{(n)}$ $(1 \leq v \leq V)$ represents the number of occurrences of word v per comment at frame t in clip n.

(b) **Comment Density:** This indicates the number of comments in each frame. Note that feature values in each clip are normalized with respect to its length and the number of comments. This feature is used to learn what temporal positions should be annotated in a target clip. Let $d_t^{(n)}$ be a feature value of frame t in clip n.

(c) **Average Length of Comments:** This indicates the average number of words in a single comment and is used to learn how long comments could be generated. In the above example with "love", the value of this feature is 2.66 ((3 + 3 + 2) / 3). Let $l_t^{(n)}$ be an average length of comments of frame t in clip n.

A set of these features is given by $\boldsymbol{o}_t^{(n)} = \{\boldsymbol{a}_t^{(n)}, \boldsymbol{w}_t^{(n)}, d_t^{(n)}, l_t^{(n)}\}$. When clip n contains T_n frames, the observable features $\boldsymbol{O}^{(n)}$ and $\boldsymbol{O}$ are given by $\boldsymbol{O}^{(n)} = \{\boldsymbol{o}_1^{(n)}, \cdots, \boldsymbol{o}_{T_n}^{(n)}\}$ and $\boldsymbol{O} = \{\boldsymbol{O}^{(1)}, \cdots, \boldsymbol{O}^{(N)}\}$.

4 MusicCommentator

MusicCommentator takes a constructive approach that tries to clarify the cognitive mechanism of humans by building and examining a computational model emulating it. As shown in Fig. 2, the system comprises a *learning* phase in which the system tries to acquire a sense of commenting (i.e., build a computational model of what comments are suitable to specific acoustic features) by experiencing many comments provided by users, and a *commenting* phase in which the model is used to generate comments suitable to the occasion. We will discuss how to design the model and then explain the two phases.

4.1 Model Formulation

Considering the characteristics of target data, we think that a reasonable model should meet the following three requirements:

1. **Joint Modeling of Acoustic and Textual Features:** When users want to produce new comments, they seem to simultaneously take into account the content of musical audio signals and the content of comments provided by other users. This suggests that well-balanced integration of them will enable the model to yield reasonable comments.

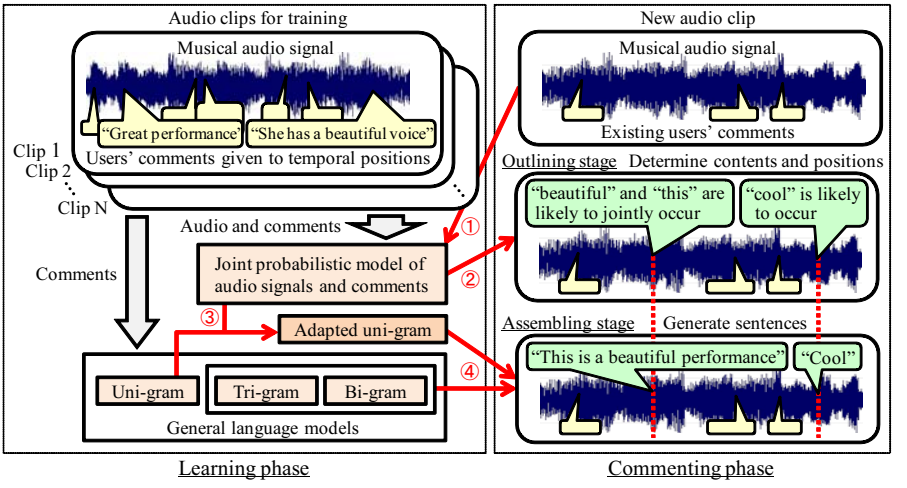

Fig. 2. Overview of MusicCommentator

2. **Temporal Modeling of Acoustic and Textual Features:** Because music is one of temporal medium, its temporal characteristics such as mood transitions should be captured by using a temporal model. Similarly, we focus on topic transitions in comments that are synchronized with music.
3. **Cross-modal Feature Binding through Temporal Contexts:** We can assume that a single latent *state*, which can be conceptually interpreted as a mood or topic, is shared behind the observable features of audio signals and comments at each frame.

To meet these requirements, we propose a joint probabilistic model of multimodal features by extending a standard hidden Markov model (HMM), as shown in Fig. 3. Let K be the number of latent states and let $\boldsymbol{z}_t^{(n)} = \{z_{t,1}^{(n)}, \cdots, z_{t,K}^{(n)}\}$ be a state representation at frame t in clip n, where $z_{t,k'}^{(n)} = 1$ and $\{z_{t,k}^{(n)} = 0 | k \neq k'\}$ if the model stays at state k' $(1 \leq k' \leq K)$. We define latent state sequences $\boldsymbol{Z}^{(n)}$ and $\boldsymbol{Z}$ as $\boldsymbol{Z}^{(n)} = \{\boldsymbol{z}_1^{(n)}, \cdots, \boldsymbol{z}_{T_n}^{(n)}\}$ and $\boldsymbol{Z} = \{\boldsymbol{Z}^{(1)}, \cdots, \boldsymbol{Z}^{(N)}\}$.

Our HMM can, like standard HMMs, be characterized by a set $\boldsymbol{\theta}$ of three kinds of parameters $\{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{\phi}\}$. $\boldsymbol{\pi}$ is a set of initial probabilities $\{\pi_1, \cdots, \pi_K\}$, where $\pi_k \equiv p(z_{1,k}^{(\cdot)} = 1)$. $\boldsymbol{A}$ is a transition matrix $\{A_{jk} | 1 \leq j, k \leq K\}$, where $A_{jk} \equiv p(z_{t,k}^{(\cdot)} = 1 | z_{t-1,j}^{(\cdot)} = 1)$. $\boldsymbol{\phi}$ is a set of parameters of output distributions that calculate the likelihoods of observable features.

Acoustic and textual features at a frame are associated with the same state. Let b_k be a joint output distribution of state k. This calculates the likelihood of $\boldsymbol{o}_t^{(n)}$, which is given by $b_k(\boldsymbol{o}_t^{(n)})$. This indicates how likely the four kinds of features $\{\boldsymbol{a}_t^{(n)}, \boldsymbol{w}_t^{(n)}, d_t^{(n)}, l_t^{(n)}\}$ jointly occur from state k. We assume that $b_k(\boldsymbol{o}_t^{(n)})$ can be decomposed into the following four likelihoods:

$$b_k(\boldsymbol{o}_t^{(n)}) = b_{a,k}(\boldsymbol{a}_t^{(n)})\ b_{w,k}(\boldsymbol{w}_t^{(n)})\ b_{d,k}(d_t^{(n)})\ b_{l,k}(l_t^{(n)}), \tag{1}$$

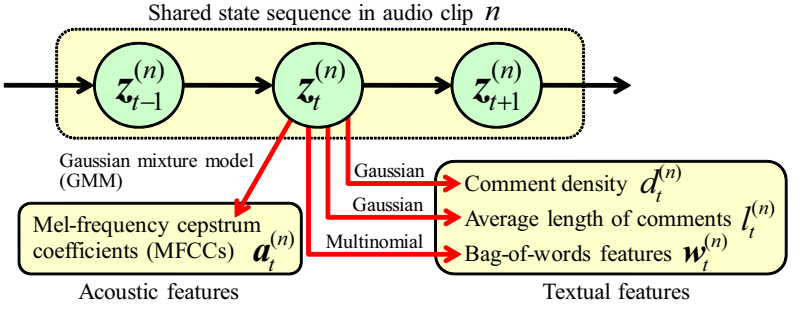

Fig. 3. Overview of our ergodic hidden Markov model

where $b_{a,k}$, $b_{w,k}$, $k_{d,k}$, and $b_{l,k}$ are designed as follows:

1. **Output Distributions of Acoustic Features:** $b_{a,k}$ is a Gaussian mixture model (GMM) of state k as in typical HMMs for speech recognition. Let M be the number of mixtures and let $g_{a,k,m}$, $\boldsymbol{\mu}_{a,k,m}$, and $\boldsymbol{\Sigma}_{a,k,m}$ be the weight, mean, and covariance of the m-th Gaussian in the GMM of state k.
2. **Output Distributions of Textual Features:** $b_{w,k}$ is a multinomial distribution for bag-of-words features. Its parameters are given by $\boldsymbol{p}_k = \{p_{k,1}, \cdots, p_{k,V}\}$. $b_{d,k}$ is a standard Gaussian representing the distribution of comment densities. Let $\mu_{d,k}$ and $\Sigma_{d,k}$ be the mean and variance of the Gaussian associated with state k. Similarly, $b_{l,k}$ is also a Gaussian for comment lengths, and its mean and variance are given by $\mu_{l,k}$ and $\Sigma_{l,k}$.

Let $\boldsymbol{\phi}_k$ be the set of parameters of output distributions of state k, given by $\boldsymbol{\phi}_k = \{\{g_{a,k,m}, \boldsymbol{\mu}_{a,k,m}, \boldsymbol{\Sigma}_{a,k,m} | 1 \leq m \leq M\}, \boldsymbol{p}_k, \mu_{d,k}, \Sigma_{d,k}, \mu_{l,k}, \Sigma_{l,k}\}$. In total, a set of all parameters of output distributions $\boldsymbol{\phi}$ is given by $\boldsymbol{\phi} = \{\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_K\}$.

Our model is an ergodic HMM, which allows any state transition at any time, because we cannot identify correct sequences of states in training data. In speech recognition, manual transcriptions of speech signals (i.e., phoneme sequences) can be directly transformed into state sequences because each phoneme is defined as a combination of several states. Left-to-Right HMMs, where state transitions are limited to match the transcriptions, are therefore commonly used in speech recognition. In contrast, we use the HMM in an unsupervised fashion.

4.2 Learning Phase

This section explains how to estimate the unknown parameters $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{\phi}\}$. Let $p(\boldsymbol{O}|\boldsymbol{\theta})$ be the likelihood of observable variables $\boldsymbol{O}$. Instead of directly maximizing the *incomplete* likelihood $p(\boldsymbol{O}|\boldsymbol{\theta})$, we try to maximize the expected *complete* likelihood of observable variables $\boldsymbol{O}$ and latent variables $\boldsymbol{Z}$ by using the Expectation-Maximization (EM) algorithm [9]. The complete likelihood is

$$p(\boldsymbol{O}, \boldsymbol{Z}|\boldsymbol{\theta}) = \prod_{n=1}^{N} p(\boldsymbol{z}_1^{(n)}|\boldsymbol{\pi}) \left[\prod_{t=2}^{T_n} p(\boldsymbol{z}_t^{(n)}|\boldsymbol{z}_{t-1}^{(n)}) \right] \prod_{t=1}^{T_n} p(\boldsymbol{o}_t^{(n)}|\boldsymbol{z}_t^{(n)}), \tag{2}$$

where $p(\boldsymbol{z}_1^{(n)}|\boldsymbol{\pi})$ is given by $\prod_{k=1}^{K} \pi_k^{z_{1,k}^{(n)}}$, which is an initial probability that clip n starts at a state specified by $\boldsymbol{z}_1^{(n)}$. We then define the Q function as follows:

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}_{old}) = \sum_{\boldsymbol{Z}} p(\boldsymbol{Z}|\boldsymbol{O}, \boldsymbol{\theta}_{old}) \log p(\boldsymbol{O}, \boldsymbol{Z}|\boldsymbol{\theta}), \tag{3}$$

where $\boldsymbol{\theta}_{old}$ is a set of the current parameters and $p(\boldsymbol{Z}|\boldsymbol{O}, \boldsymbol{\theta}_{old})$ is a posterior probability of latent variables $\boldsymbol{Z}$. $Q(\boldsymbol{\theta}|\boldsymbol{\theta}_{old})$ indicates the *expected* complete log-likelihood of all variables $\boldsymbol{O}$ and $\boldsymbol{Z}$ when we regard $\boldsymbol{\theta}$ as a variable of the function. Thus, the optimized parameters are obtained by maximizing $Q(\boldsymbol{\theta}|\boldsymbol{\theta}_{old})$ and are then set to $\boldsymbol{\theta}_{old}$ next time. This is iterated until $Q(\boldsymbol{\theta}|\boldsymbol{\theta}_{old})$ converges.

In the E-step of the EM algorithm, the objective is to calculate the posterior distribution $p(\boldsymbol{Z}|\boldsymbol{O}, \boldsymbol{\theta}_{old})$. For convenience, we define some new symbols as:

$$\gamma(\boldsymbol{z}_t^{(n)}) \equiv p(\boldsymbol{z}_t^{(n)}|\boldsymbol{O}, \boldsymbol{\theta}_{old}),\ \xi(\boldsymbol{z}_{t-1}^{(n)}, \boldsymbol{z}_t^{(n)}) \equiv p(\boldsymbol{z}_{t-1}^{(n)}, \boldsymbol{z}_t^{(n)}|\boldsymbol{O}, \boldsymbol{\theta}_{old}), \tag{4}$$

$$\gamma(\boldsymbol{y}_{t,k}^{(n)}) \equiv p(\boldsymbol{y}_{t,k}^{(n)}|\boldsymbol{O}, \boldsymbol{\theta}_{old}) = p(\boldsymbol{y}_{t,k}^{(n)}|\boldsymbol{z}_t^{(n)})\gamma(\boldsymbol{z}_t^{(n)}), \tag{5}$$

where $\gamma(\boldsymbol{z}_t^{(n)})$ is a posterior distribution of latent variable $\boldsymbol{z}_t^{(n)}$. $\xi(\boldsymbol{z}_{t-1}^{(n)}, \boldsymbol{z}_t^{(n)})$ is a joint posterior distribution of adjacent latent variables $\boldsymbol{z}_{t-1}^{(n)}$ and $\boldsymbol{z}_t^{(n)}$. For each t, $\gamma(\boldsymbol{z}_t^{(n)})$ consists of K probabilities that sum up to unity. $\xi(\boldsymbol{z}_{t-1}^{(n)}, \boldsymbol{z}_t^{(n)})$ is expressed as a $K \times K$ probability matrix whose elements sum up to unity. Let $\gamma(z_{t,k}^{(n)})$ be the conditional probability of $z_{t,k}^{(n)} = 1$ and let $\xi(z_{t-1,j}^{(n)}, z_{t,k}^{(n)})$ be that of $z_{t-1,j}^{(n)} = z_{t,k}^{(n)} = 1$, given $\boldsymbol{O}$ and $\boldsymbol{\theta}_{old}$. These probabilities can be efficiently calculated by using the forward-backward algorithm [11]. $\boldsymbol{y}_{t,k}^{(n)}$ is a vectorial variable, $\{y_{t,k,1}^{(n)}, \cdots, y_{t,k,M}^{(n)}\}$. This shows which Gaussian is responsible for generating $\boldsymbol{a}_t^{(n)}$ among M Gaussians in GMM $b_{a,k}$, where $y_{t,k,m'}^{(n)} = 1$ and $\{y_{t,k,m}^{(n)} = 0 | m \neq m'\}$ when the m'-th Gaussian is responsible. $p(\boldsymbol{y}_{t,k}^{(n)}|\boldsymbol{z}_t^{(n)})$ is expressed as a $K \times M$ probability matrix that reserves the responsibilities of KM Gaussians of GMMs $\{b_{a,1}, \cdots, b_{a,K}\}$ for observation $\boldsymbol{a}_t^{(n)}$.

In the M-step, we try to maximize the Q function $Q(\boldsymbol{\theta}|\boldsymbol{\theta}_{old})$. Substituting Eqn. (2) for Eqn. (3), we get

$$\begin{aligned} Q(\boldsymbol{\theta}|\boldsymbol{\theta}_{old}) = \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma(z_{1,k}^{(n)}) \log \pi_k + \sum_{n=1}^{N}\sum_{t=2}^{T_n}\sum_{j=1}^{K}\sum_{k=1}^{K} \xi(z_{t-1,j}^{(n)}, z_{t,k}^{(n)}) \log A_{jk} \\ + \sum_{n=1}^{N}\sum_{t=1}^{T_n}\sum_{k=1}^{K} \gamma(z_{t,k}^{(n)}) \log p(\boldsymbol{o}_t^{(n)}|\boldsymbol{\phi}_k), \end{aligned} \tag{6}$$

where the last term can be decomposed into four terms as $\log p(\boldsymbol{o}_t^{(n)}|\boldsymbol{\phi}_k) = \log b_{a,k}(\boldsymbol{a}_t^{(n)}) + \log b_{w,k}(\boldsymbol{w}_t^{(n)}) + \log b_{d,k}(d_t^{(n)}) + \log b_{l,k}(l_t^{(n)})$ We can thus independently update the parameters of four kinds of distributions (GMM $b_{a,k}$, multinomial distribution $b_{w,k}$, and two Gaussians $b_{d,k}$ and $b_{l,k}$) by using the Lagrange multiplier method. We get the updating formula as follows:

$$\pi_k = \frac{\sum_{n=1}^{N} \gamma(z_{1,k}^{(n)})}{\sum_{n=1}^{N}\sum_{k=1}^{K} \gamma(z_{1,k}^{(n)})},\ A_{jk} = \frac{\sum_{n=1}^{N}\sum_{t=2}^{T_n} \xi(z_{t-1,j}^{(n)}, z_{t,k}^{(n)})}{\sum_{n=1}^{N}\sum_{l=1}^{K}\sum_{t=2}^{T_n} \xi(z_{t-1,j}^{(n)}, z_{t,l}^{(n)})},$$

$$g_{a,k,m} = \frac{\sum_{n=1}^{N}\sum_{t=1}^{T_n} \gamma(y_{t,k,m}^{(n)})}{\sum_{n=1}^{N}\sum_{t=1}^{T_n}\sum_{m=1}^{M} \gamma(y_{t,k,m}^{(n)})},\ \boldsymbol{\mu}_{a,k,m} = \frac{\sum_{n=1}^{N}\sum_{t=1}^{T_n} \gamma(y_{t,k,m}^{(n)}) \boldsymbol{a}_t^{(n)}}{\sum_{n=1}^{N}\sum_{t=1}^{T_n} \gamma(y_{t,k,m}^{(n)})},$$

$$\boldsymbol{\Sigma}_{a,k,m} = \frac{\sum_{n=1}^{N}\sum_{t=1}^{T_n} \gamma(y_{t,k,m}^{(n)})(\boldsymbol{a}_t^{(n)} - \boldsymbol{\mu}_{a,k,m})^2}{\sum_{n=1}^{N}\sum_{t=1}^{T_n} \gamma(y_{t,k,m}^{(n)})},\ \boldsymbol{p}_k = \frac{\sum_{n=1}^{N}\sum_{t=1}^{T_n} \gamma(z_{t,k}^{(n)}) \boldsymbol{w}_t^{(n)}}{\sum_{n=1}^{N}\sum_{t=1}^{T_n} \gamma(z_{t,k}^{(n)})},$$

$$\mu_{d,k} = \frac{\sum_{n=1}^{N}\sum_{t=1}^{T_n} \gamma(z_{t,k}^{(n)}) d_t^{(n)}}{\sum_{n=1}^{N}\sum_{t=1}^{T_n} \gamma(z_{t,k}^{(n)})},\ \Sigma_{d,k} = \frac{\sum_{n=1}^{N}\sum_{t=1}^{T_n} \gamma(z_{t,k}^{(n)})(d_t^{(n)} - \mu_{d,k})^2}{\sum_{n=1}^{N}\sum_{t=1}^{T_n} \gamma(z_{t,k}^{(n)})},$$

$$\mu_{l,k} = \frac{\sum_{n=1}^{N}\sum_{t=1}^{T_n} \gamma(z_{t,k}^{(n)}) l_t^{(n)}}{\sum_{n=1}^{N}\sum_{t=1}^{T_n} \gamma(z_{t,k}^{(n)})},\ \Sigma_{l,k} = \frac{\sum_{n=1}^{N}\sum_{t=1}^{T_n} \gamma(z_{t,k}^{(n)})(l_t^{(n)} - \mu_{l,k})^2}{\sum_{n=1}^{N}\sum_{t=1}^{T_n} \gamma(z_{t,k}^{(n)})}. \quad (7)$$

4.3 Commenting Phase

This section explains how to provide comments suitable to a target audio clip. Like the training data, the audio signal and provided comments are characterized by a sequence of acoustic features, $\boldsymbol{a}' = \{\boldsymbol{a}'_1, \cdots, \boldsymbol{a}'_{T'}\}$, three sequences of textual features, $\boldsymbol{w}' = \{\boldsymbol{w}'_1, \cdots, \boldsymbol{w}'_{T'}\}$, $\boldsymbol{d}' = \{d'_1, \cdots, d'_{T'}\}$, and $\boldsymbol{l}' = \{l'_1, \cdots, l'_{T'}\}$, where T' is the number of frames. This phase consists of an *outlining* stage and an *assembling* stage. The latter estimates how many comments and what content should be provided at each frame. The former concatenates a suitable number of words in an appropriate order by taking language constraints into account.

Outlining Stage. We first determine a most likely sequence of latent states in the target, $\boldsymbol{z}' = \{\boldsymbol{z}'_1, \cdots, \boldsymbol{z}'_{T'}\}$, with the Viterbi algorithm [10]. When $z'_{t,k}$ is 1 at frame t ($1 \leq t \leq T'$), the most likely density there, $\hat{d}_t$, is given by the mode (most likely observation) of the Gaussian $b_{d,k}$, i.e., mean $\mu_{d,k}$. From the density distribution over the entire clip, we can determine how many comments should be provided in each frame. Similarly, when $z'_{t,k}$ is 1, we can get most likely bag-of-words features (occurrence probabilities of significant words) $\hat{\boldsymbol{w}}_{\boldsymbol{t}}$ to be $\boldsymbol{p}_k$.

We here cannot generate sentences that are appropriate as natural language, i.e., reasonable sequences of words, because bag-of-words features only outlines the content of comments. Therefore, we should solve the following problems:

1. We do not have occurrence probabilities of non-significant words such as conjunctions and auxiliary verbs, which are indispensable for natural language.
2. We do not have individual occurrence probabilities of inflectional words that have the same base morpheme within them (see 2a in Section 3).

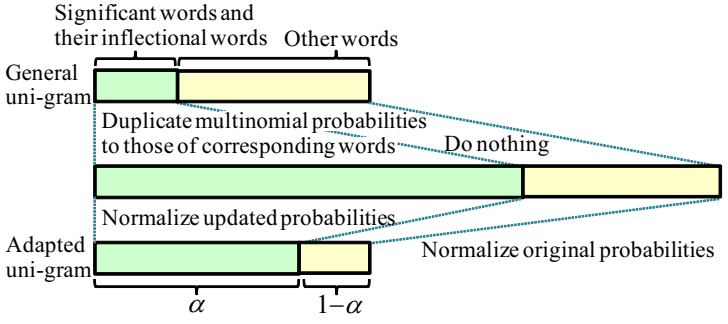

Fig. 4. Adaptation of general uni-gram to multinomial distribution

3. We cannot determine an appropriate order of words because the current model does not take into account sequential relations between words.

For example, suppose that two words "it" and "love" are highly likely to occur and the comment length is likely to be three. We cannot synthesize a comment like "It is loved" or "I love it" because the probabilities of "is" and "loved" are not given and we therefore do not know which sentence is more appropriate.

Assembling Stage. To solve the three problems described above, we propose a comment generation method based on adaptation of general language models (uni-, bi-, and tri-grams) that are learned from numerous comments of all clips in the training data. Unlike what we do in the learning phase, we distinguish between different words that have the same base morpheme (e.g., we distinguish "took" from "taken"'). The uni-gram can be used for solving the first and second problems, and the bi- and tri- grams contribute to solving the third one.

Fig. 4 shows a sketch of how the probabilities $\hat{\boldsymbol{w}}_{\boldsymbol{t}}$ of significant words at frame t are incorporated into the general uni-gram, which includes all the words appearing in the training data. We duplicate the probability of each significant word to those of its inflectional words that have different surface expressions. For example, the probabilities of words "took" and "taken" are set to be the same as that of word "take." Then, because the sum the probabilities of significant words and their inflectional ones is greater than 1, the probabilities are scaled so that their sum is α, which is a control parameter given in advance. On the other hand, the probabilities of other words containing non-significant words and their inflectional ones are scaled so that their sum is $1 - \alpha$. As a result, we now get the adapted uni-gram (a set of occurrence probabilities of *all* words).

When $z'_{t,k}$ is 1 at frame t, we assume that a most likely comment (word sequence) $\hat{\boldsymbol{c}}_t$ and a most likely comment length (number of words) $\hat{l}_t$ should be determined according to the following generative model:

$$\{\hat{\boldsymbol{c}}_t, \hat{l}_t\} = \underset{\boldsymbol{c},l}{\operatorname{argmax}}\, p(\boldsymbol{c}, l; \boldsymbol{\theta}_k) = \underset{\boldsymbol{c},l}{\operatorname{argmax}}\, p(\boldsymbol{c}|l; \boldsymbol{\theta}_k) p(l; \boldsymbol{\theta}_k), \tag{8}$$

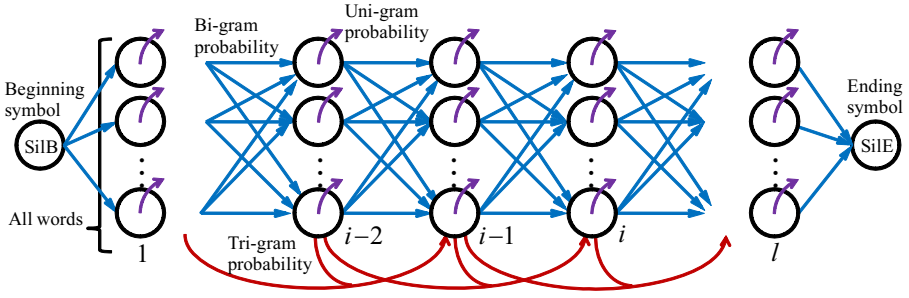

Fig. 5. Probability calculation on word trellis

where $p(l;\boldsymbol{\theta}_k)$ is a likelihood that a comment generated from state k consists of l words. Its value is calculated according to Gaussian $b_{l,k}$ and $p(\boldsymbol{c}|l;\boldsymbol{\theta}_k)$ is a conditional probability that comment $\boldsymbol{c}$ is generated when its length is given by l. Note that for readability we hereafter omit the estimated parameter $\boldsymbol{\theta}_k$. To get $\hat{\boldsymbol{c}}$, we have only to calculate $\mathrm{argmax}_{\boldsymbol{c}}\, p(\boldsymbol{c}|l)$ for each length.

To estimate $\mathrm{argmax}_{\boldsymbol{c}}\, p(\boldsymbol{c}|l)$, we propose a method that can find a most likely path of words on a trellis including all words by using the Viterbi algorithm [10]. As shown in Fig. 5, each node corresponds to a specific word and the observation probabilities of words in each column are given by the adapted uni-gram. Transition probabilities between nodes are determined as bi- and tri-grams. We let SilB and SilE be special symbols (silent words) that indicate the beginning and ending of comments. The likelihood of comment $\boldsymbol{c}$ is given by

$$p(\boldsymbol{c}|l) = p(w_1|\mathrm{SilB}) \left(\prod_{i=2}^{l} p(w_i|w_{i-2}, w_{i-1}) \right) p(\mathrm{SilE}|w_{l-1}, w_l), \tag{9}$$

where w_i is the i-th word in comment $\boldsymbol{c}$ and w_0 is SilB. $p(w_i|w_{i-2}, w_{i-1})$ is an *adapted* trigram probability, which is calculated with linear interpolation of the general tri- and bi- grams p_t, p_b and the adapted uni-grams p'_u as follows: $p(w_i|w_{i-2}, w_{i-1}) \leftarrow \beta_t p_t(w_i|w_{i-2}, w_{i-1}) + \beta_b p_b(w_i|w_{i-1}) + \beta_u p'_u(w_i)$, where β_t, β_b, and β_u are weighting factors of the tri-, bi-, and uni-grams.

5 Evaluation

We experimentally evaluated how accurately the system predicted comments that are freely provided on arbitrary temporal positions by users.

5.1 Conditions

The audio clips (tracks) we used were included in the video clips taken from the music category of the video sharing service *Nico Nico Douga*. Specifically, we focus on music performances whose titles included *"Ensoushitemita"* ("We/I played

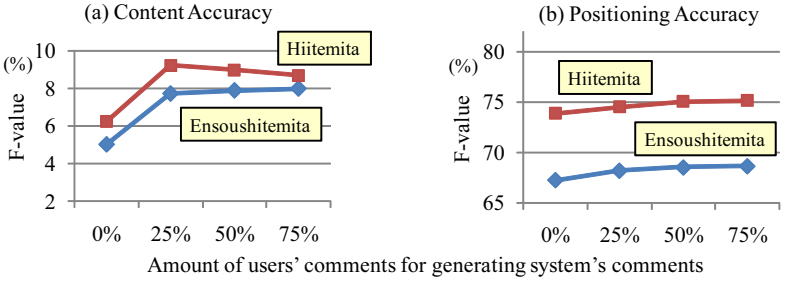

Fig. 6. Results of experimental comment generation

something, not limited to musical instruments, e.g., music box and wooden gong"), or *"Hiitemita"* ("We/I played piano or stringed instruments, e.g., violin and guitar"). Some of them were performed by multiple people, e.g., sessions, bands, or ensembles. There are many popular clips that follow these conventional naming rules in the title. We collected the most popular 100 *"Ensoushitemita"*-category clips that are shorter than 10 minutes according the the number of comments that roughly reflects its popularity. Then, the first 1100 comments, which were available as many as possible in all the clips, were extracted. Note that the first 1100 comments do not mean the 1100 comments taken from the beginning within each clip, but mean the 1100 comments taken from the beginning of its submission to the video sharing service. As for the *"Hiitemita"* category having more comments, we were able to extract 2400 comments from each of 100 clips. The control parameters were set as $V = 2082, 2278$, $K = 200$, $M = 8$, $\alpha = 0.9$, and $\beta_t = \beta_b = \beta_u = 1.0$ by trial and error.

The experiments were conducted in the way of 4-fold cross fold validation. First, all audio clips with provided comments were randomly divided into four groups. Three groups were used as a training set in the learning phase and the other group was used as a test set in the commenting phase. Switching the choice of test set, we conducted four trials. 4-fold cross fold validation was furthermore performed in each trial by dividing the provided comments of each test clip into four groups. To estimate a most likely sequence of states in a given clip, the system used either no comments (i.e., only acoustic features) or one, two, or three groups (i.e., 0%, 25%, 50%, or 75%) of comments on the clip. That is, we tested four settings. The remaining comments was used as ground truth.

To evaluate the results, we calculated the word-based F-value, which is given by $F = \frac{2PR}{P+R}$, where P and R are the precision and recall rates. We focused on each word of the system-generated comments. A word in a system's comment was considered reasonable if it appeared in users' comments annotated in the neighborhood of the system's comment. The error tolerance was set to 5 seconds.

$$P = \frac{\#\text{appropriate words}}{\#\text{words of system's comments}}, \quad R = \frac{\#\text{appropriate words}}{\#\text{words of users' comments}}, \tag{10}$$

5.2 Results

As shown in Fig. 6(a), the F-values could be improved even if only 25% of users' comments of a target clip was available for adding new comments. Although the F-values reached at most 10%, we think these results were promisingly reasonable because it is impossible to completely predict what comments are provided by users at a word level even for humans. Note that when we evaluated only the temporal positions and lengths of generated comments (i.e., allowed errors in word selection), the F-values were around 70%, as shown in Fig. 6(b). One may say that it is enough to list most likely words as a rough suggestion. However, we believe that sentences of natural language are much better in terms of readability although they are often grammatically or semantically strange because n-grams cannot all inter-word dependencies contained in a sentence.

The F-values were not furthermore improved when we increased the amount of users' comments over 25% for adding new comments. This indicates that the current system cannot deal with widely diverse comments. That is, the model cannot create various comments that are essentially different from each other in their meanings once a specific state is determined for given acoustic and textual features. Comments freely provided by humans without constraints are widely diverse. A major reason that the F-values for the *"Ensoushitemita"* category were lower than those for the *"Hiitemita"* category could be the wider diversity of the comments on former clips. The title *"Hiitemita"* means limited kinds of instruments such as piano and guitar were used in the video.

We also found that the current system is not always useful because general comments like "it is very cool" and "great" tend to be generated. The F-values of frequently used positive words such as "cool" or "great" were around 40%. This was closely related to the limitation of the statistical approach. If we can use a huge amount of users' comments for training the HMM (we actually used over 100,000 comments), the probabilistic model tries to capture universal characteristics of the data. However, it is not appropriate to spoil the diversity of humans' comments for our task. We should tackle this problem in the future.

6 Conclusion

We presented MusicCommentator that generates comments (short sentences of natural language) and provide them at appropriate temporal positions. The system is based on a multi-modal HMM that associates acoustic features with textual ones through latent sequences of states. These sequences correspond to temporal transitions of both musical moods and comment topics. To estimate the parameters of the HMM, we used a likelihood maximization method so that many examples of how users have provided comments can be well explained with the model. Given a new audio clip, the system concatenates suitable words in an appropriate order by using general language models.

The experimental results were promising but revealed that we are still far from the ultimate goal of building a computer that can express the impressions of video clips as natural language as humans do. Because commenting is one of the

most sophisticated cognitive functions of humans, it would be hard to precisely emulate even if we use the state-of-the-art techniques of machine learning. We think, however, that our study is an important first challenge. We plan to improve MusicCommentator by incorporating advanced methods of recognizing musical content such as rhythm and melody. This kind of multi-aspect modeling could help the system generate comments that are more appropriate and diverse.

Acknowledgement. This study was partially supported by CREST, JST.

References

1. Ames, M., Naaman, M.: Why We Tag: Motivations for Annotation in Mobile and Online Media. In: ACM CHI, pp. 971–980 (2007)
2. Nakamura, S., Shimizu, M., Tanaka, K.: Can Social Annotation Support Users in Evaluating the Trustworthiness of Video Clips? In: ACM WICOW (2008)
3. Hamasaki, M., Takeda, H., Nishimura, T.: Network Analysis of Massively Collaborative Creation of Multimedia Contents –Case Study of Hatsune Miku videos on Nico Nico Douga–. In: uxTV, pp. 165–168 (2008)
4. Whitman, B., Rifkin, R.: Musical Query-by-Description as a Multiclass Learning Problem. In: IEEE MMSP 2002, pp. 153–156 (2002)
5. Turnbull, D., Barrington, L., Torres, D., Lanckriet, G.: Semantic Annotation and Retrieval of Music and Sound Effects. IEEE Trans. on ASLP 16(2), 467–476 (2008)
6. Bertin-Mahieux, T., Eck, D., Maillet, F., Lamere, P.: Autotagger: A Model for Predicting Social Tags from Acoustic Features on Large Music Databases. J. of New Music Research (JNMR) 37(2), 115–135 (2008)
7. Tzanetakis, G., Cook, P.: Musical Genre Classification of Audio Signals. IEEE Trans. on SAP 10(5), 293–302 (2002)
8. Kudo, T., Yamamoto, T., Matsumoto, Y.: Applying Conditional Random Fields to Japanese Morphological Analysis. In: EMNLP (2004)
9. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. J. R. Stat. Soc (B) 39(1), 1–38 (1977)
10. Forney, G.D.: The Viterbi Algorithm. IEEE 61(3), 268–278 (1973)
11. Baum, L.E., Petrie, T., Soules, G., Weiss, N.: A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. Annals of Mathematical Statistics 41(1), 164–171 (1970)