Toward a chanting robot for interactively teaching English to children

Ryo Nagata[1], *Tomoya Mizumoto*[2], *Kotaro Funakoshi*[3] *Mikio Nakano*[3]

[1]Faculty of Intelligence and Informatics, Konan University, Japan
[2]Graduate School of Information Science, Nara Institute of Science and Technology, Japan
[3]Honda Research Institute Japan Co., Ltd., Japan
rnagata @ konan-u.ac.jp., tomoya-m @ is.naist.jp, {funakoshi,nakano} @ jp.honda-ri.com

Abstract

To acquire a second language, one must develop an ear and tongue for the correct stress and intonation patterns of that language. In English education, there is a rhythmic teaching method called Jazz Chants. This paper proposes a new application for second language education which combines Jazz Chants with a companion robot, and reports our technical investigations toward realizing such a robot. Investigated were two key technologies: predicting stresses in Jazz Chants and synthesizing chant speech. Experiments show promising results and reveal requirements for further improvement.

Index Terms: stress-timed rhythm, teacher assist, intelligent tutoring system, human-robot interaction

1. Introduction

To acquire a spoken language, one must develop an ear and tongue for the correct stress and intonation patterns of the spoken language. This is normally difficult for those who are acquiring a second language whose sound system is not similar to that of their first language. An example pair would be English and Japanese of which sound systems are quite different.

In English language teaching, there is an effective method called *Jazz Chants*[1] for working on the sound system. A chant is the rhythmic expression of natural language which links the rhythms of spoken American English to the rhythms of traditional American jazz — the rhythm, stress and intonation pattern of what children would hear from an educated native speaker in natural conversation. In chants, each stressed word is pronounced with an equal duration (i.e., isochronism) often with physical activities such as clapping or jumping. To support this, stressed words are sometimes (but not always) marked with the asterisk * or underlines in teaching material for chants (Hereafter, teaching material for chants will be referred to as *chant text*). An example chant text would be as follows:

```
  *      *      *          *
Frank, Hank, walk to the bank.
 *    *    *         *
Jill, Phil, run up the hill.
```

Teachers and children can read them out, putting stresses on the marked words.

According to the textbook [1] for Jazz chants, the use of chants has the following three advantages in language learning and teaching:

1. Acquiring stress and intonation patterns
2. Memorizing everyday phrases
3. Learning grammar and vocabulary

Since chants require only sound and movement to teach, they are especially suitable for children who are yet familiar with written language.

Chants are also well-suited to computer-assisted language learning. Specifically, the authors have been trying to develop a chanting robot that interactively teaches English based on chants. Basically, the robot reads English sentences out following the sound system of chants with a certain motion such as clapping and jumping so that the leaner can read out along with the robot. It may make special motions such as stepping and turning round as rewards for the learner if he or she pronounces the given English sentences well. It may also interact with the learner by addressing and responding to him or her (e.g., "Robot: *What's your name?* Learner: *My name is Sue. What's your name?* Robot: *My name is Robo.*"). Alternatively, he or she can teach the robot using chants instead of directly learning from the robot, that is, learning by teaching; he or she teaches the robot English by reading chants out, and the robot gradually develops an ear for the correct stress and intonation patterns of English if he or she reads out well. This results in further interactions between the learner and the robot.

To develop a chanting robot, there are at least six technical challenges to overcome:

1. Predicting stresses in chants
2. Synthesizing the speech of chants
3. Producing the movements of the robot
4. Recognizing learner's utterance
5. Understanding learner's utterance
6. Generating chanting phrases

The first is a prediction task to determine whether each word gets stressed or not in a given text. This is used when the robot reads out a chant text with motion. The second is the process for synthesizing the speech of chants based on stress prediction. The third determines the movement of the robot including stepping, clapping, and turning as teachers of English do in classroom. Such movement attracts learners' attention and enhance learning effects. The fourth to sixth are somewhat optional in a basic chanting robot, but necessary to develop a fully interactive one. They are used for addressing and responding to the learner, including the evaluation of leaner's utterance. It might seem to be too difficult to implement. However, given that the target learners are children who have not acquired a large English vocabulary nor complicated syntactic structures, we believe that there are a lot we can do even with the existing techniques. For example, the robot may be able to recognize the learner's utterance and respond to it by replacing one of the words as in "Learner: *I like dogs.* Robot: *I like cats*".

[1] Jazz Chants® is a registered trademark of Oxford University Press. In this paper, Jazz Chants will be simply referred to as *chants*.

This paper explores methods for achieving the first and second. Although our ultimate goal is to develop a chanting robot as just explained, the methods themselves are useful for teachers. Chant texts often do not indicate which word gets stressed since native speakers of English have no difficulty in determining it. By contrast, teachers who are non-native speakers of English have often difficulties in determining and recognizing stresses[2]. The methods can be used to visualize where to put stresses and to demonstrate how chants sound when teachers use chants.

In order to predict stresses in chants, one could apply conventional pitch-accent prediction methods such as [2, 3]. However, although stresses in chants share similar properties with pitch accents, they seem not to be identical. Stresses in a chant text basically satisfies the constraint that the number of stresses in a chant text is divisible by four so that they can be read out with music. Related to this, chants are more isochronism-oriented as shown by the above example chant text than pitch accents. It is likely that one will have to modify the conventional pitch-accent prediction methods to achieve a good performance in stress prediction in chants.

Considering these differences, this paper investigates how well simple methods work on stress prediction and speech synthesis for chants. Then, this paper discusses the results to reveal technical challenges left in the simple methods.

The rest of this paper is structured as follows. Section 2 describes the method for predicting stresses for chants. Section 3 describes and discusses an attempt to synthesize a chant speech by using a ready-made speech synthesizer.

2. Predicting stresses in chants

2.1. Method

Before getting into the details of the method, let us first observe an example sentence in the textbook [1]:

*   *   *   *
Frank, Hank, walk to the bank.

This example implies the hypothesis that content words such as *Frank* and *walk* tend to get stressed and function words such as *to* and *the* do not. Also note that the example can be equivalently expressed as:

Frank/S ,/N Hank/S ,/N walk/S to/N the/N bank/S ./N

where S and N denote stress and not-stress, respectively. In this, a careful reader may notice its resemblance to the part-of-speech (POS) tagging problem as in:

Frank/NN ,/, Hank/NN ,/, walk/VB to/PP the/DT bank/NN ./.

where NN, VB, PP, and DT denote noun, verb, preposition, and determiner, respectively.

Having observed this, one might come to think of solving the stress prediction problem as the POS tagging problem. This is an overarching idea of the method.

Since a Hidden Markov Model (HMM) has been shown to be effective in the POS tagging problem, the proposed method uses HMMs to predict stresses. To be precise, the paper proposes two HMM-based methods. The first one is based on an HMM whose input and output are words and stress tags (i.e., S or N), respectively; this method will be referred to as HMM-Word, hereafter. An example of the input and output of HMM-Word would be as:

Input: Frank, Hank, walk to the bank.
Output: Frank/S ,/N Hank/S ,/N walk/S to/N
the/N bank/S ./N

The HMM is trained on chant texts annotated with stresses.

The other takes POSs as the input instead of words; this method will be referred to as HMM-POS, hereafter. Each sentence is first POS-tagged by an existing tool. Then, the POS tags are put into the HMM to obtain stress tags. Namely, the input and output corresponds to the POSs and stress tags. Finally, the stress tags are merged with the input sentence. To illustrate HMM-POS, let us consider the previous example, again:

Frank, Hank, walk to the bank.

This would be POS-tagged as:

Frank/NN ,/, Hank/NN ,/, walk/VB
to/PP the/DT bank/NN ./.

Then, the POS tags:

NN , NN , VB PP DT NN .

are put into the HMM, giving the result:

NN/S ,/N NN/S ,/N VB/S PP/N DT/N

Finally, it is merged with the input sentence:

Frank/S ,/N Hank/S ,/N walk/S to/N
the/N bank/S ./N

The HMM is trained in the same manner as HMM-Word.

It should be noted that the above methods do not predict which syllable gets stressed. However, it should not be so problematic since it can be solved by looking it up in a dictionary. Thus, this paper focuses only on predicting which word gets stressed.

2.2. Evaluation and discussion

For evaluation, we used 71 chant texts, which were annotated with stresses, in the textbook[1]. In all, the 71 chant texts consisted of 657 sentences and 2,473 words.

We implemented the proposed methods using trigram-based HMMs with the interpolation of unigrams and bigrams. We used an in-house POS-tagger which had 44 POS tags based on the Penn Treebank tag set for HMM-POS. To measure the performance, we used recall, precision, F-measure, and accuracy. All measures were calculated by leave-one-out cross-validation (one text was left out each time).

For comparison, we implemented two other methods as baselines in addition to the proposed methods. In the first baseline, all words are tagged as S (Baseline1). In the second baseline, all content words are tagged as S (Baseline2).

Table 1 summarizes the experimental results. Table 1 reveals that HMM-Word achieved a similar performance to that of HMM-POS, which we had not expected. A possible reason for this is that since we used only one textbook and the leave-one-out method in the evaluation, the words were not so diverse, which helped HMM-Word perform well. Of course, unknown words did appear and affected the performance of HMM-Word. Proper nouns were especially problematic in HMM-Word. By contrast, HMM-POS tended to make correct predictions for proper nouns because they were correctly tagged as proper nouns even if they were unknown words.

[2] For instance, those who are not teachers of English but of other subjects are in charge of English language teaching in primary schools in Japan. Historically, there had been no English classrooms in primary schools in Japan.

Table 1: Experimental results

Method	R	P	F	A
Baseline1	**1.00**	0.633	0.797	0.633
Baseline2	0.760	0.772	0.766	0.692
HMM-Word	0.852	**0.861**	0.856	0.810
HMM-POS	0.895	0.840	**0.866**	**0.817**

R: Recall, P: Precision, F: F-measure, A: Accuracy

At the same time, we found that words were more effective than POSs in some cases. For example, both *you* and *it* were tagged as PRP by the POS tagger. However, *you* tended to have a stress more often than *it*. This example suggests that some POS tags are too coarse. In other words, certain words such as *you* and *it* should be treated as separate POSs to achieve better performance. Alternatively, information on both words and POSs can be considered in prediction by using a certain kind of classifier such as conditional random field (CRF) as in the method [2].

Contrary to this, it turned out that the POS tag set used in the evaluation was sometimes too fine for our task. For example, there were four types of noun POS (singular noun; plural noun; singular proper noun; plural proper noun) in the POS tagger. The differences in these nouns are not so crucial for determining stresses for chants. Considering this, it is likely that collapsing some classes of POSs into one will improve the performance of HMM-POS. To sum up, while the information obtained from POSs is effective in predicting where to stress, it is necessary to determine the adequate granularity of the tag set.

The results showed that simple methods for predicting stresses in chants worked fairly well. At the same time they revealed some defects common to all the methods. One of them is that chants basically satisfy the constraint that the number of stresses in a chant is divisible by four as explained in Section 1. The proposed methods obviously do not consider the constraint. Rhymes are also informative but not considered (e.g., Fran*k*, Han*k*, wal*k* to the ban*k*). Another is that stresses are determined by information beyond the sentence. For instance, whether the previous sentence is interrogative or not sometimes determines where to stress as in *Where is the book? It is ON the table*. Note that the preposition *on* is stressed while normally prepositions are not stressed. This kind of global information may be able to be handled by using other classifiers such as decision lists and CRF. We expect more improvement by considering these in the proposed method.

3. Synthesizing chants

One of the initial goals for building a chanting robot is to automatically produce a naturally sounding chant speech from a chant text. For that purpose, the robot can predict stress positions by using the method proposed in section 2, but it also has to manage vocalizing timing so that the predicted stress positions align with equally-timed beats. Therefore, simple use of a Text-To-Speech (TTS) system is not enough for the chanting robot.

In this section, as a first step for such a fully automatic synthesis of chant speech, we report our attempt to manually synthesize the speech for a chant in a textbook by using the Festival text-to-speech system[3]. In what follows, we use the chant text "Two, four, six, eight." [1]:

```
  *     *     *    *
Two, four, six, eight.
          *           *               *              *
I don't want to be late. I don't want to be late.
  *     *     *         *
Two, four, six, eight, ten.
  *        *         *       *
Say it again. Say it again.
```

3.1. Festival TTS for singing

The Festival TTS has a function for singing. Given an XML data such as shown in Figure 1, Festival synthesizes a singing speech. The XML data in Figure 1 is for the chant "Two, four, six, eight". We made the XML data from the time information obtained from an example performance described in the next subsection.

```
<!DOCTYPE SINGING PUBLIC "-//SINGING//DTD SINGING mark up//EN"
 "Singing.v0_1.dtd" []>
<SINGING BPM="75">
<REST BEATS="0.10"></REST>
<DURATION BEATS="0.42"><PITCH NOTE="C3">two</PITCH></DURATION>
<REST BEATS="0.46"></REST>
<DURATION BEATS="0.65"><PITCH NOTE="C3">four</PITCH></DURATION>
<REST BEATS="0.26"></REST>
<DURATION BEATS="0.81"><PITCH NOTE="C3">six</PITCH></DURATION>
<REST BEATS="0.47"></REST>
<DURATION BEATS="0.38"><PITCH NOTE="C3">eight</PITCH></DURATION>
<REST BEATS="0.10"></REST>
<DURATION BEATS="0.20"><PITCH NOTE="C3">I</PITCH></DURATION>
<DURATION BEATS="0.17"><PITCH NOTE="C3">don’t</PITCH></DURATION>
<DURATION BEATS="0.45,0.19"><PITCH NOTE="C3,C3">wanna</PITCH></DURATION>
<DURATION BEATS="0.20"><PITCH NOTE="C3">be</PITCH></DURATION>
<DURATION BEATS="0.36"><PITCH NOTE="C3">late</PITCH></DURATION>
<REST BEATS="0.46"></REST>
<DURATION BEATS="0.20"><PITCH NOTE="C3">I</PITCH></DURATION>
<DURATION BEATS="0.20"><PITCH NOTE="C3">don’t</PITCH></DURATION>
<DURATION BEATS="0.38,0.27"><PITCH NOTE="C3,C3">wanna</PITCH></DURATION>
<DURATION BEATS="0.14"><PITCH NOTE="C3">be</PITCH></DURATION>
<DURATION BEATS="0.49"><PITCH NOTE="C3">late</PITCH></DURATION>
<REST BEATS="0.78"></REST>
<DURATION BEATS="0.40"><PITCH NOTE="C3">two</PITCH></DURATION>
<REST BEATS="0.45"></REST>
<DURATION BEATS="0.63"><PITCH NOTE="C3">four</PITCH></DURATION>
<REST BEATS="0.38"></REST>
<DURATION BEATS="0.72"><PITCH NOTE="C3">six</PITCH></DURATION>
<DURATION BEATS="0.38"><PITCH NOTE="C3">eight</PITCH></DURATION>
<REST BEATS="0.00"></REST>
<DURATION BEATS="0.59"><PITCH NOTE="C3">ten</PITCH></DURATION>
<REST BEATS="0.29"></REST>
<DURATION BEATS="0.54"><PITCH NOTE="C3">say</PITCH></DURATION>
<DURATION BEATS="0.26"><PITCH NOTE="C3">it</PITCH></DURATION>
<DURATION BEATS="0.15,0.61"><PITCH NOTE="C3,C3">again</PITCH></DURATION>
<REST BEATS="0.42"></REST>
<DURATION BEATS="0.60"><PITCH NOTE="C3">say</PITCH></DURATION>
<DURATION BEATS="0.23"><PITCH NOTE="C3">it</PITCH></DURATION>
<DURATION BEATS="0.17,0.71"><PITCH NOTE="C3,C3">again</PITCH></DURATION>
</SINGING>
```

Figure 1: *Festival singing XML for "Two, four, six, eight."*

One can specify the length and note (pitch) of each syllable and rest (pause). A length is specified relative to a beat, that is, the length of 0.5 means it spans half of the beat interval. By adjusting these durations appropriately, stress positions will be aligned with beats.

3.2. Data and annotation

An example performance recorded in the CD appended to the textbook [1] was annotated manually by using the Audacity software[4]. Figure 2 shows the waveform and syllable-level annotation on this example performance[5]. The signal found in rest intervals is the sound of piano accompaniment. The figure also shows annotated piano sounds, where a label "2" means a single chord of piano spanning 2 beat intervals. The piano annotation confirms that not the onsets (i.e., consonants) of syllables but the nucleuses (i.e., vowels) are aligned with beats as pointed out in past literature.

[3] http://www.cstr.ed.ac.uk/projects/festival/

[4] http://audacity.sourceforge.net/

[5] "want to" was pronounced as "wanna" in the performance.

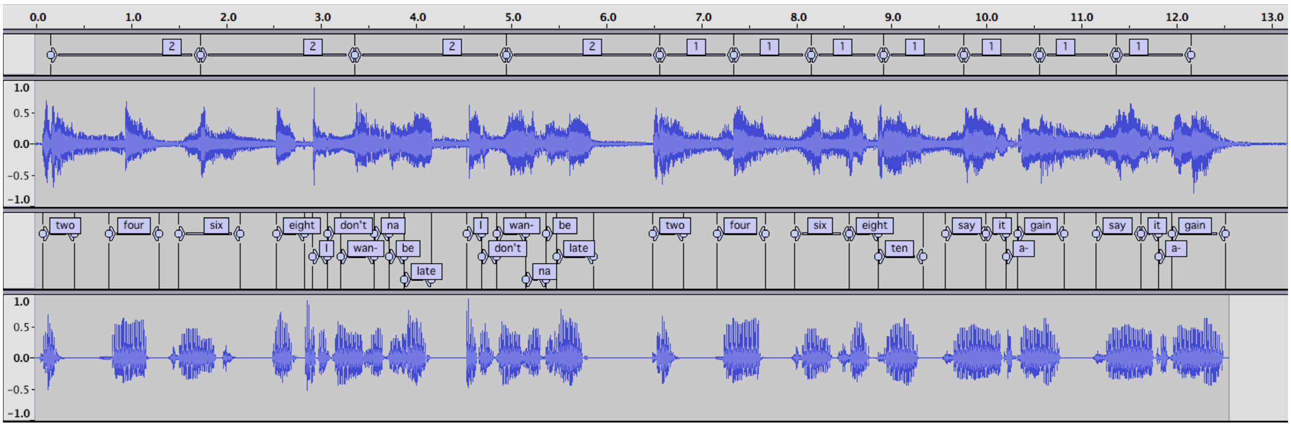

Figure 2: *An example chant performance (second track) annotated with syllables (third track) and piano sounds (first track), and a synthesized chant speech (bottom track).*

3.3. Synthesis results and discussion

The time information of the performance was extracted from the annotation and used to generate the XML data shown in Figure 1. The mean beat interval length was 0.799 sec ($SD = 0.022$, about 75 BPM) which was computed from the piano annotation.

Figure 2 shows the comparison between the original performance and the synthesized speech. The mean time difference of syllable onsets between them is -0.030 sec ($SD = 0.036$, the origin is the original performance). Those of the first half and the second half are -0.046 ($SD = 0.038$) and -0.010 ($SD = 0.020$), respectively. Although the synthesized speech tends to start a little bit earlier than the original (especially in the first half), the two speech sounds are synchronized fairly well.

Using Festival for a chanting robot seems promising. Yet, exploring other synthesis tools such as VOCALOID[6] is also future work.

In the trial described above, all the necessary time information is obtained from an example performance. If, however, the robot creates/modifies a chant by itself, it also has to generate time information by itself. We will tackle this problem as the next step of this work.

Two gaps between the original performance and synthesized speech are observable in Figure 2. First, the synthesized speech is very flat in terms of intensity while the original performance is more dynamic (Compare the waveforms for the word *four*). Second, some non-stressed syllables almost disappeared in the original performance but they are recognizable in the synthesized speech (e.g., the first syllable "a-" of the word *again*). These gaps should be filled in so as to produce a more naturally sounding chant speech.

For simplicity, we ignored the pitches of syllables (all the notes in the XML are C3). This apparently makes the synthesized chant sound amusia. Chants are not songs and they do not have specific melodies, but totally flat chanting sounds unnatural. Adding appropriate subtle melodies is desired. At least, we have to add appropriate pitch accent or intonational emphasis on semantically important words.

4. Conclusions

This paper proposed a new application for second language education which combines *Jazz Chants* with a conversational companion robot, that is, *a chanting robot*, and reported our technical investigations toward realizing it. Investigated were two key technologies: predicting stresses in chants and synthesizing chant speech, and we obtained promising results and grasped issues to be tackled. Obviously, many challenges which are necessary to achieve interactive teaching by a robot (at least challenges 3. to 6. described in Section 1) are still left untouched.

While there are many video-based teaching materials and tutoring systems adopting animated agents called embodied conversational agents (ECAs), using a robot has several advantages. That is, it will provide a more vivid experience, will be able to teach groups in any formation, will not be choosy about places for teaching, and will be able to approach and encourage students on its initiative (Video materials and ECAs have to wait for students to come).

Moreover, L2 training robots will be a good research platform or test bed for human-robot interaction or multimodal spoken dialogue systems because it will not require difficult spoken language understanding and users will continuously interact with them spontaneously, which enables researchers to collect huge data and to test their technologies over a longer span.

5. Acknowledgments

We would like to thank Chikako Arima and Dr. Tatsuya Kitamura for their useful comments on this work.

6. References

[1] Graham, C., Creating Chants and Songs, Oxford, 2006.

[2] Gregory, M.L. and Altun A., "Using Conditional Random Fields to Predict Pitch Accents in Conversational Speech," Proc. of 42nd Annual Meeting on Association for Computational Linguistics, pp.47–54, July 2004.

[3] Margolis, A. and Ostendorf, M., "Acoustic-based Pitch-accent Detection in Speech: Dependence on Word Identity and Insensitivity to Variations in Word Usage," Proc. of 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, pp.4514–4516, Apr. 2009.

[6] http://www.vocaloid.com/en/index.html