Alignment of Sound Track with Text in a TV Drama

Seigo Tanimura, Hiroshi Nakagawa
Information Technology Center, The University of Tokyo.
7-3-1 Hongo, Bunkyo, Tokyo, JAPAN 113-0033
{tanimura,nakagawa}@r.dl.itc.u-tokyo.ac.jp

Abstract

We propose a system to align a sound track and a part of TV drama video contents with a script. We first use the number of moras in each sentence of speech line, the sounding time in a sound track and the shot change time in the motion image to align them approximately. Then we perform DP matching to align a sequence of words obtained from a speech recognition system applied to a sound track with each sentence of speech lines in a script. Confident correspondences obtained from the DP matching of the words act as the pivots to improve alignment accuracy iteratively. The results show that around a half of the sentences in a script were aligned within the differences of up to two seconds.

1 Introduction

Alignment of video to text is essential to make video contents flexibly reusable. Although it seems to be promising for this purpose to apply a speech recognition technology to the sound track of a video content, speech recognition for a TV drama or a feature film is actually difficult. The major obstacle significant in a drama and film is the poor accuracy of speech recognition. Due to the background music, the noise from the environment in a location and audio compression like MPEG, the accuracy of speech recognition is only around 10-30%. Thus the result of speech recognition to the sound track of a drama or film does not have enough quality to directly reuse the video contents.

Although the result of speech recognition is of no use by itself, we still have the script of a drama or film. Hence alternative method is to align the script of a drama or film to the video. Several proporsals have been made to solve this problem. Yaginuma et al. (Yaginuma and Sakauchi, 1996) proposed time alignment of a TV drama by the physical shot changes in video, the volume in the sound track and the number of characters in the speech lines of the script. However, the accuracy of alignment by their method achieved only 70% for a sentence due to lack of speech recognition in their method.

While speech recognition can improve the accuracy of alignment, we need to solve a couple of problems to apply speech recognition to a sound track of a drama or film. A state-of-the-art speech recognition system performs speech recognition in a sentence-to-sentence manner. In addition to that, applying speech recognition directly to a whole length of sound track, say 30 minutes or longer, involves unrealistic costs in both time and memory. Hence a sound track needs to be divided into sentences prior to applying a speech recognition system. Due to the large number of sentences in a TV drama, it may result in inferior accuracy to divide a whole sound track into sentences. We can avoid this by dividing a sound track roughly first into, say, logical scenes. Then we apply more precise segmentation based on the sentences to the logical scenes.

We developed a system to align each sentence of speech lines in a script written in Japanese to the corresponding part of a sound track accompanying the script spoken in Japanese. Our proposing system consists of six modules. Figure 1 shows the architecture of our system. Note that "x alignment" means "alignment of sequences formed from x" in the rest of this paper.

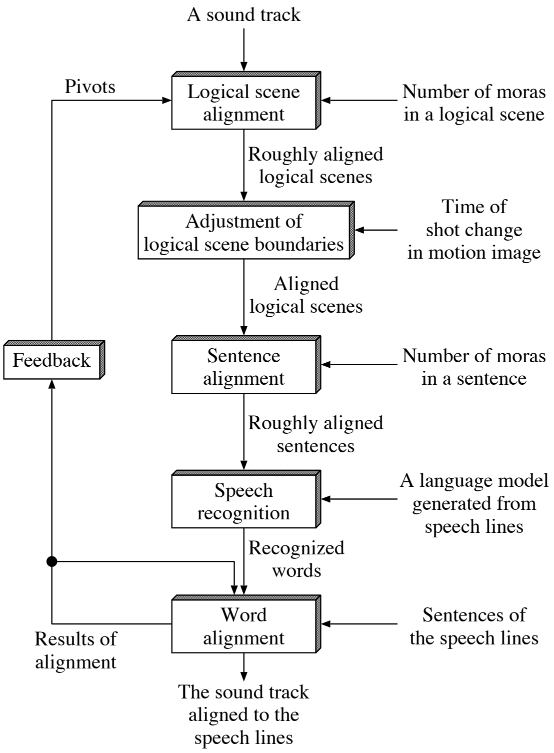
A sound track
Pivots
Logical scene alignment
Number of moras in a logical scene
Roughly aligned logical scenes
Adjustment of logical scene boundaries
Time of shot change in motion image
Aligned logical scenes
Feedback
Sentence alignment
Number of moras in a sentence
Roughly aligned sentences
Speech recognition
A language model generated from speech lines
Recognized words
Word alignment
Sentences of the speech lines
Results of alignment
The sound track aligned to the speech lines

Figure 1: The architecture of our system

In section 2, we describe logical scene alignment and logical scene boundary adjustment. The sentence alignment module is discussed in section 3. We describe the speech recognition module in 4. Section 5 details the word alignment module. The mechanism of feedback is discussed in section 6. We present our experimental results in section 7 and conclude in section 8.

2 Logical Scene Alignment

The word alignment module in our system is capable of aligning a scene of only up to around several thousand words, or several minutes. A scene longer than, say, 10 minutes cannot be aligned directly by word-to-word manner with a practical computational cost. Hence we need to develop other means of approximate alignment capable of handling a scene of 10-30 minutes.

A logical scene in a script can be an alternate of a word to align a sound track to a script. A logical scene has the following features:

- A logical scene can be extracted from a script as easily as a word. A typical logical scene consists of the scene number and the title, followed by the speech lines and directions.
- A logical scene in a motion image often begins and ends at a shot change. It can be detected by the shot detection system which usually uses the change of color histograms between a couple of consecutive frames in a motion image.

These facts imply that a logical scene can be extracted both from a script and motion image. Hence we adopt a logical scene as an alternative of a word to align a sound track to the script. The problem is that a logical scene may consist of not a single but several shots. Thus duration of a logical scene needs to be estimated using the speech lines of a script.

It is well known in Japanese linguistics and phonetics that a length of utterance duration in Japanese is basically proportional to the number of the moras in the uttered word (Kubozono, 1999). Counting this fact, we seek the starting and ending time of a part of the sound track which corresponds to a logical scene in the script as the first approximation, so that the duration of sounding segment corresponding to every logical scene of the script gets proportional to the number of moras in the logical scene. The starting and ending time are then adjusted to the nearest shot change time in the motion image. The sounding segments in the sound track are extracted by comparing power of every 25msec period in the sound track to the predetermined threshold. The number of moras in a logical scene is computed by the pronunciation of the words in the logical scene obtained from a morphological analyzer. We apply a commercial scene cut tool (Hitachi, Ltd., 1997) to detect the shot changes in the motion image.

3 Sentence Alignment

A logical scene generally consists of several sentences. However, unlike extracting the logical scenes in a whole drama, it is difficult to extract the sentences directly from a sound track or a sequence of image frames. To segment out each sentence in the sound track, we apply the method to use the numbers of moras and the duration of sounding segments described in section 2, except that we seek a part of the sound track corresponding to not a logical scene but a sentence. Since an utterance of a speech line may not begin or end at a shot change, we modify the alignment method proposed in section 2 by omitting adjustment of starting and ending time to the nearest shot change time.

4 Speech Recognition

4.1 Language and Acoustic Models

In our system, the words to be uttered are given as the speech lines of a script in advance. A word bigram language model used during speech recognition is generated from the whole speech lines to improve the accuracy of a speech recognition system. Every word in a language model is uttered at least once, and no other words are uttered with this tailored language model. This avoids recognizing words not appearing in the speech lines.

It also brings about a better accuracy in speech recognition to choose the acoustic model adapted for a speaker. However, a general method of speaker adaptation has some major problems in our system, that is, the speakers in the segments of the sound track are not given as the input. Furthermore, not only one but several speakers may utter in a single segment of the sound track. Thus we cannot determine a suitable acoustic model prior to speech recognition. In order to solve this problem, we perform speech recognition with multiple acoustic models (Ming et al., 1999), say female and male models, because the difference of these two models affects the accuracy of a speech recognition system significantly. To perform speech recognition simply, these acoustic models are used in parallel. This allows us to improve the accuracy of sentence alignment without misselecting a suitable acoustic model, described in section 5.2.

4.2 A Speech Recognition System and Filtering out Noisy Words

We use JULIUS (Ito et al., 1998) as a speech recognition system. A language model is generated by applying a Japanese morphological analyser JUMAN (Kurohashi and Nagao, 1997) and CMU-Cambridge SLM Toolkit (Clarkson, 1997) to the speech lines. We use HMMs of 16 mixed density for triphones of 3000 states as the acoustic models. HMMs for female and male speakers are used in parallel.

We postprocess the recognized words in order to improve the accuracy of sentence alignment. A speech recognition system treats unuttering duration in a sound track as a comma or a full stop. However, these do not usually match to the commas and full stops in the speech lines. Thus commas and full stops in the recognized words and speech lines are apparently noise in word alignment. We filter out these noisy words from the recognized words and the words in the speech lines prior to word alignment.

5 Word Alignment with DP Matching

In this module, we align for each of logical scenes the sequences of the recognized words to the sentences of the speech lines based on the similarity between a pair of words. The similarity between words involved in this alignment is computed by performing mora-based DP matching. More precisely, our word alignment system consists of two level alignment modules; a word alignment module for a pair of sentences described in section 5.2 and a mora alignment for a pair of words described in section 5.1. To proceed the word alignment for sentences, the word alignment module invokes the mora alignment module for words interactively.

5.1 Mora Alignment for Words

We describe in this section our mora-based DP matching. Let A_m be a sequence of moras of a word in a sentence of a speech line in the script, and B_m be a recognized word of the speech line, respectively. They are defined as follows(A subscript of m stands for a mora):

$$\begin{aligned} A_m &= \{a_{m1}, a_{m2}, \ldots, a_{mi}, \ldots, a_{mI}\} \quad (1) \\ B_m &= \{b_{m1}, b_{m2}, \ldots, b_{mj}, \ldots, b_{mJ}\} \quad (2) \end{aligned}$$

where a_{mi} is the ith mora in a word of a sentence, b_{mj} is the jth mora in a recognized word, I is the number of moras in the word of a sentence, and J is the number of the moras in a recognized word. Then we define a similarity between a pair of moras, $s_m(a_{mi}, b_{mj})$ as follows:

$$s_m(a_{mi}, b_{mj}) = \begin{cases} 3 & (a_{mi} = b_{mj}) \\ 2 & \left(\begin{array}{l} \text{Only the vowel of } a_{mi} \text{ is} \\ \text{equal to the vowel of } b_{mj} \end{array} \right) \\ 0 & (\text{None of the above}) \end{cases} \quad (3)$$

A vowel in a recognized word is more confident than a consonant in general. Thus we give a similarity to a pair of moras with the identical vowel and different consonants as well. Using the expression (3), we iterate an expression (6) with the initial conditions of expressions (4) and (5) as follows to compute $g_m(a_{mI}, b_{mJ})$:

$$g_m(a_{m1}, b_{m1}) = 0 \quad (4)$$

$$g_m(a_{mi}, b_{m1}) = g_m(a_{m1}, b_{mj}) = -\infty \quad (5)$$

$$g_m(a_{mi}, b_{mj}) = \max_{q=1,2,\ldots,p-1} \begin{cases} g_m(a_{mi-q-1}, b_{mj-1}) + s_m(a_{mi-q}, b_{mj}) \\ g_m(a_{mi-1}, b_{mj-1}) + s_m(a_{mi}, b_{mj}) \\ g_m(a_{mi-1}, b_{mj-q-1}) + s_m(a_{mi}, b_{mj-q}) \\ \quad + \sum_{r=1}^{q} s_m(a_{mi}, b_{mj-q+r}) \end{cases} \quad (6)$$

where p is a parameter to forbid stretching and shrinking p or more moras locally.

Figure 2 shows the local constraint and the weights in the expression (6). The black dots show that the alignment paths can grow from these dots to (i, j). The numbers accompanied by the paths are the weights of them. This constraint assures that the possible maximum of $g_m(a_{mI_m}, b_{mJ_m})$ does not depend on J_m. We iterate the expression (6) until $g_m(a_{mI_m}, b_{mJ_m})$ gets computed. Finally, the similarity between A_m and B_m is defined as follows:

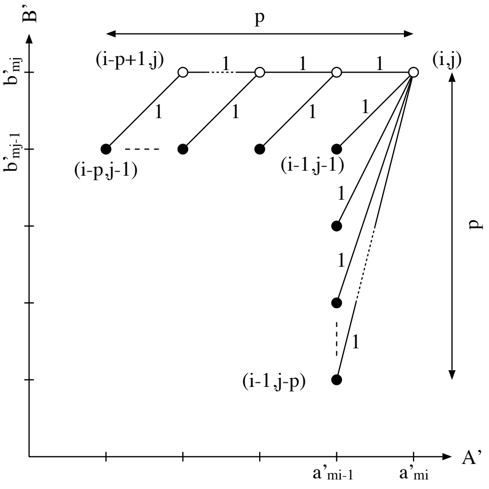

Figure 2: The local constraint and the weights of the paths in mora alignment

$$S_m(A_m, B_m) = \begin{cases} g_m(a_{mI}, b_{mJ}) + \frac{10}{|I-J|+1} & (A_m \neq B_m) \\ \left(g_m(a_{mI}, b_{mJ}) + \frac{10}{|I-J|+1} \right)^2 & (A_m = B_m) \end{cases} \quad (7)$$

The second term in the expression (7) is included to discourage matching words with excess difference between the numbers of the moras. If $A_m = B_m$, i.e. they have the identical pronunciation with each other, we square $g_m(a_{mI_m}, b_{mJ_m}) + \frac{10}{|I_m - J_m|+1}$ to give higher similarity to these words corresponding to each other. We use $S_m(A_m, B_m)$ in word alignment for sentences, described in section 5.2.

5.2 Word Alignment for Sentences

Word alignment algorithm for sentences is also DP matching as well as the mora alignment for words is. Let C_w and D_w be a sequence of words in a sentence and a recognized word of the speech line, respectively. They are defined as follows(A subscript of w

stands for a word):

$$C_w = \{c_{w1}, c_{w2}, \ldots, c_{wi}, \ldots, c_{wI}\} \quad (8)$$

$$D_w = \{d_{w1}, d_{w2}, \ldots, d_{wj}, \ldots, d_{wJ}\} \quad (9)$$

where c_{wi} is the ith word in the whole sentence of the speech line, d_{wj} is the jth word in the recognized words, I is the number of words in the whole sentences, and J is the number of the recognized words. Then we define a similarity between a pair of words, $s_w(c_{wi}, d_{wj})$ using the results of the mora alignment module for words described in section 5.1 as follows:

$$s_w(c_{wi}, d_{wj}) = S_m(A_{mi}, B_{mj}) \qquad (10)$$

where A_{mi} and B_{mj} are the sequences of moras in the words c_{wi} and d_{wj} respectively, and $S_m(A_{mi}, B_{mj})$ is defined in the expression (7). Using the expression (10), we iterate an expression (13) with the initial conditions of expressions (11) and (12) as follows to compute $g_w(c_{wI}, d_{wJ})$:

$$g_w(c_{w1}, d_{w1}) = 0 \qquad (11)$$

$$g_w(c_{wi}, d_{w1}) = g(c_{w1}, d_{wj}) = -\infty \qquad (12)$$

$$g_w(c_{wi}, d_{wj}) = \max_{q=1,2,\ldots,p-1} \begin{cases} g_w(c_{wi-q-1}, d_{wj-1}) + 2s_w(c_{wi-q}, d_{wj}) \\ \quad + \sum_{r=1}^{q} s_w(c_{si-q+r}, d_{wj}) \\ g_w(c_{wi-1}, d_{wj-1}) + 2s_w(c_{wi}, d_{wj}) \\ g_w(c_{wi-1}, d_{wj-q-1}) + 2s_w(c_{wi}, d_{wj-q}) \\ \quad + \sum_{r=1}^{q} s_w(c_{wi}, d_{wj-q+r}) \end{cases} \qquad (13)$$

where p is a parameter to forbid stretching and shrinking p or more words locally.

Figure 3 shows the local constraint and the weight of the paths. The black dots show that the alignment paths can grow from these dots to (i, j). The numbers accompanied by the paths are the weights of them. The expressions (11) and (12) ensure that the first words in the sequences always correspond to each other. We iterate the expression (13) until $g_w(c_{wI}, d_{wJ})$ gets computed.

5.3 Selection of the actually uttered words

As mentioned in section 4, we obtain two sequences of recognized words with female and male acoustic models respectively. In order to select the actually uttered words dynamically between these two sequences, we developed a method to select the appropriate sequence upon finding a word d_{i-1} uttered prior to d_i.

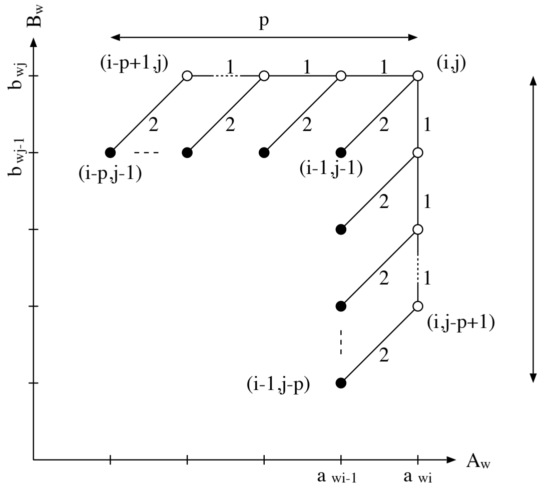

Figure 3: The local constraint and the weights of the paths in DP matching of the word sequences

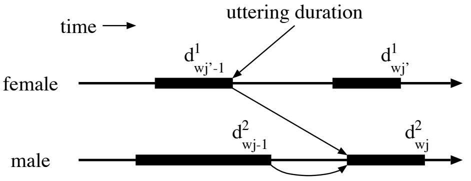

Figure 4: Selection of words uttered prior to d^2_{wj}

Figure 4 shows our method to select the words uttered prior to d^2_{wj} from the words recognized by acoustic models for female and male. We select from each of the word sequences the words with the nearest ending time to the beginning time of d^2_{wj} from all of the words with the ending times prior to the beginning time of d^2_{wj}. For example, we find $d^1_{wj'-1}$ and d^2_{wj-1} as the words uttered prior to d^2_{wj} in figure 4. Then each of $d^1_{wj'-1}$ and d^2_{wj-1} is substituted into d_{wj-1} to compute $g_w\left(c_{wi}, d^2_{wj}\right)$. Then the word that gives the highest $g_w\left(c_{wi}, d^2_{wj}\right)$ is selected as d_{wj-1}.

6 Improvement by Feedback

We improve the accuracy of alignment by fixing confident correspondences in the result of the word alignment module discussed in section 5 as fixed pivots in logical scene alignment.

A confident correspondence should not consist of short words. Short words in our system refer to the words of only one or two moras. Most of such the words are functional words in Japanese. They appear quite frequently in any speech lines. Moreover, a speech recognition system may misrecognize utterances or even noises to end up with spurious functional words. On the contrary, a correspondence of long words with an identical pronunciation can be confident. Such a correspondence shows that the utterance is recognized correctly with a matching word in the speech lines. Counting these facts, we define a confident corresponding word pair as follows:

- The corresponding words have an identical pronunciation.
- The pronunciation should have a length of at least three moras.

We pick up the correspondences satisfying both of these conditions shown above from the result of word alignment as the confident correspondences. Using these confident correspondences as fixed pivots, we realign the sounding segments of sound track with each logical scenes and sentences of the speech lines, and reperform the speech recognition and word alignment described in section 2-5, respectively.

7 Experimental Results

We evaluated the alignment accuracy of our system experimentally. Table 1 shows the sample scene for our experiment.

We first counted for each cycle of iteration the number of the recognized words with three or more moras and the number of the aligned sentences including at least one pivot. The results are shown in figure 2 and 3, respectively.

Table 1: Sample scene

Number of logical scenes	24
Number of sentences	91
Number of words with three or more moras	502
Duration of the sound track [min:sec]	14:44

Table 2: The numbers of the recognized words with three or more moras

Iteration	No. of words
Cycle 1	59
Cycle 2	57
Cycle 3	59

Table 3: The numbers of the aligned sentences with at least one pivot

Iteration	No. of sentences
Cycle 1	31
Cycle 2	37
Cycle 3	36

Although the numbers of recognized words shown in table 2 do not cover all of the words in the script, we can still approximate the accuracy of speech recognition by these results. The accuracy of speech recognition stayed around 12%, indicating a poor quality of the sample sound track. Nevertheless a third of the sentences were aligned with pivots. In addition, we gained quite a few number of newly aligned sentences as iteration proceeds, as shown in table 3. These results imply that the pivots in a sentence obtained in the first cycle diffuse to the neighbour sentences.

In order to investigate the effect of the pivot gain shown in table 3, we evaluated the accuracy of alignment by the following method. We measured the difference between the utterance beginning/ending time of the recognized word aligned to the first/last word of each sentence and the correct utterance beginning/ending time of the sentence, which is expressed as ϵ_b/ϵ_e henceforth. We then counted the number of the sentences satisfying $|\epsilon| \leq E$ where ϵ is either ϵ_b or ϵ_e and E is one of 1, 3 or 5 seconds. The average of $|\epsilon|$ for the whole sentences, Av. was also computed. The results are shown in table 4.

Table 4: The numbers of the sentences satisfying $|\epsilon| \leq E$ and the average of $|\epsilon|$

Iteration	ϵ	E			Av.[s]
		1[s]	3[s]	5[s]	
Cycle 1	ϵ_b	17	31	42	16.9
	ϵ_e	7	21	35	18.3
Cycle 2	ϵ_b	32	47	60	8.4
	ϵ_e	16	36	53	10.1
Cycle 3	ϵ_b	31	55	62	9.9
	ϵ_e	19	43	53	12.0

We can state from these results that our alignment system can align not only a sentence recognized correctly but also its neighbor sentences. On the other hand, the average of $|\epsilon|$ did not increase in cycle 3 because the level of noise was extremely higher than the level of utterance for 21 sentences uttered in a running train. Due to the poor accuracy of speech recognition of these 21 sentences, we obtained only 5 pivots at most from these sentences, ending up with the large Av.s shown in table 4.

8 Conclusion

We proposed a system to align a sound track and a sequence of image frames with sentences of speech lines in a TV drama. Our next target is to improve the accuracy of speech recognition and to seek a promising application area of our alignment method.

Acknowledgement

This work is supported in part by the Grant-in-Aid for Creative Basic Research: Multi Media Mediation Mechanism(09NP1401), of the Ministry of Education, Science, Sports and Culture, Japan.

References

Philip Clarkson. 1997. The CMU-Cambridge statistical language modeling toolkit v2. `http://svr-www.eng.cam.ac.uk/~prc14/toolkit.html`.

Hitachi, Ltd. 1997. Mediachef/CUT for Windows 95.

Katsunobu Ito, Tatsuya Kawahara, Kazuya Takeda, and Kiyohiro Shikano. 1998. Japanese dictation toolkit. *Proc. of the 12th Annual Conference of Japanese Society for Artificial Intelligence.* (In Japanese).

Haruo Kubozono. 1999. *Nihongo no Onsei(Phonetics in Japanese).* Iwanami Shoten Publishers. (In Japanese).

Sadao Kurohashi and Makoto Nagao, 1997. *Japanese Morphological Analysis System JUMAN Ver. 3.4.* (In Japanese).

Ji Ming, Philip Hanna, Darryl Stewart, Marie Ownes, and F. Jack Smith. 1999. Improving speech recognition performance by using multi-model approaches. *ICASSP 99*, 1:161–164.

Y. Yaginuma and M. Sakauchi. 1996. Content-based drama editing based on intermedia synchronization. *Proc. of the IEEE Computer Society, International Conference on Multimedia Computing and Systems '96*, pages 322–329, 6.