Algorithms for estimation of comic speakers considering reading order of frames and texts

Omori, Yuga
Graduate School of Information Science and Electrical Engineering, Kyushu University

Nagamizo, Kota
ForeVision Inc.

Ikeda, Daisuke
Faculty of Information Science and Electrical Engineering, Kyushu University

https://hdl.handle.net/2324/6781038

出版情報：2022 12th International Congress on Advanced Applied Informatics IIAI-AAI 2022 Proceedings, pp.367-372, 2022-07. IEEE
バージョン：
権利関係：

Algorithms for estimation of comic speakers considering reading order of frames and texts

Yuga Omori
Graduate School of Information Science and Electrical Engineering
Kyushu University
Fukuoka, Japan
omori.yuga.960@s.kyushu-u.ac.jp

Kota Nagamizo
ForeVision Inc.
Tokyo, Japan
kota_nagamizo@forev.co.jp

Daisuke Ikeda
Faculty of Information Science and Electrical Engineering
Kyushu University
Fukuoka, Japan
ikeda.daisuke.899@m.kyushu-u.ac.jp

***Abstract*—Machine learning methods in recent years have focused on multimodal input and cross-modal tasks, and they are used as approaches to problems in various domains. Associating comic texts and characters using these approaches is informative for commercial activities such as speech synthesis and automatic translation of texts. In this study, we address the task of associating a text with a speaker in comics. It is challenging to correspond between them because these are not self-evidently attached, and few studies have attempted. These previous studies have less considered the continuity of comics such as narrative flow or contextual information. We assume that considering the continuity of comics is effective for speaker estimation. This paper proposes algorithms for estimating the reading order of frames or texts, and it also proposes methods for estimating speakers based on these orders. As a result, our proposed method improves accuracy compared to previous methods. Consideration of the frame order is an effective clue to the comic speaker estimation.**

***Index Terms*—Speaker estimation, Comic, Multimodal**

I. Introduction

The digital comics market continues to grow year after year. In particular, the estimated sales of the Japanese digital market in 2021 is 411.4 billion yen (approximately 4 billion dollars), that is 20.3% higher than in that in 2020 [1]. According to [1], this is because of the stay-at-home demand due to the coronavirus disease 2019 and the appearance of vertical-scrolling comics for smartphones. If these digital comics are equipped with text-to-speech and automatic translation, they will add value not found in traditional paper comics.

A first candidate to achieve them is machine learning. In recent years, many machine learning methods have proposed and used for problems in various domains. For examples, several models based on Transformer [2] have resulted good accuracy on many problems, regardless of the domain, such as natural language or images. Moreover, these models are also used for multimodal media such as video. In recent years, there has been research on cross modal tasks, such as the Tacotron2 [3], which outputs an audio that reads an input text.

Comic is a multimodal medium which has images, such as characters, backgrounds, and texts of utterance or narration. The above methods are expected to be applied to text-to-speech and automatic translation, which are classified as cross-modal tasks. For these applications, it is required automatic understanding through association between texts and characters. However, such association is not obvious because there are some minor differences in the use of comic objects, such as frames and word bubbles, in each comic. Therefore, it is challenging to estimate comic texts' speakers automatically.

Few studies have attempted this task because there exist limited datasets about comics. In addition, these studies do not adequately consider continuity, such as narrative flow or context. Since comic is a medium that represents continuous narrative scenes, we hypothesize that the narrative flow is effective for speaker estimation. In this study, we contribute to estimate comic texts' speakers as follows: First, we define reading orders of frames and texts to use the continuity, and propose algorithms to automatically estimate them. Second, we propose three speaker estimation methods: one follows the frame order, another one follows the text order, and the other one partially modifies a previous method.

II. Related work

In this section, we briefly see two previous studies on automatic estimation of comic texts' speakers.

Rigaud *et al.* [4] used eBDtheque dataset [5] and non public dataset composed by the first volume of a Japanese comic. The eBDtheque dataset contains information annotated frames, characters, text lines, word bubbles (in this dataset, it is called speech balloon). In [4], they employed the Euclidean distance between a target text and a character, and between a bubble tail and a character. [4] suggested that Euclidean distance between a tail and a character is important for speaker estimation.

Abe *et al.* [6], [7] constructed Manga109 dataset [8], [9] and used. This dataset uses 109 comics drawn by professional Japanese manga artists. This dataset has annotated metadata as follows: (1) frames; (2) text regions of utterance, monologue, and narration; and (3) characters' bodies and faces. Note that word bubbles are not included. Although Manga109 has no ground-truth data for each text to estimate speakers, they also constructed the ground-truth dataset for speaker estimation using Manga109 and used [6], [7]. Abe *et al.*, similarly to Rigaud *et al.*, used the Euclidean distance between a text and a character and found the following information to be useful for this estimation: (i) Whether a target text and a character are

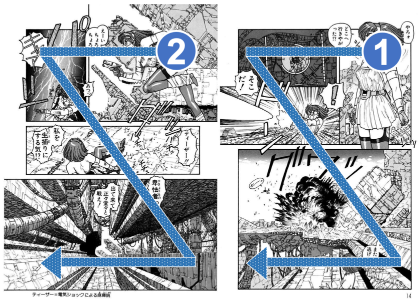

Fig. 1. The general trajectory of reading order in Japanese manga. (C)ARMS/Masaki Kato [8], [9]

placed in the same frame; and (ii) First-person pronouns and Sentence-ending particles in texts. In particular, they found that (i) contributes significantly to the estimation.

The main points of the previous studies mentioned above are as follows: closeness between a text and a character is a clue to speaker estimation; and the frames have closeness that is different from Euclidean distance.

III. Algorithms

The comics are media to reproduce continuity of the real world by various things, such as continuous frames. In this paper, we try to test our hypothesis that such continuous things in comics would help to estimate the speaker. Here we focus on the order of frames or texts among such continuous things in comics.

Therefore, we need the order of them. However, limited comic dataset has such information as metadata. So, we introduce algorithms to put orders on frames and texts in this section. Hereafter, these algorithms will be referred to as Ordering_Frames and Ordering_Texts. First, we consider Ordering_Frames, and then Ordering_Texts based on the Ordering_Frames. Finally, we evaluate the proposed algorithms.

We assume that any object in a comic is surrounded by a rectangle and the object has four (x, y) coordinates of the rectangle as Manga109, where the top-left of a page is the origin, its right-hand side area is positive x-axis, and its downside one is positive y-axis. That is, the objects, such as frame and text region, have xmax, xmin, ymax, and ymin of four values to represent its rectangle. Henceforth, text in this section essentially refers to a region with one or more text lines.

A. *Ordering_Frames*

We believe that the reading order of a comic depends on the starting position to read and the text orientation. For example, in Japanese manga, the starting position is the top right corner because we start reading vertical texts from the top right corner. From the top right corner, we move to the left top, then to the right bottom, and finally to the left bottom of the single page. The trajectory of this movement looks like horizontally flipped "Z" as shown in Fig. 1.

Algorithm 1 shows a pseudo-code of our algorithm to put frames in order. The main function of this algorithm is a recursive function ALIGN. Given a set of comic objects such as frames or texts, it sorts out them in reading order.

We use functions SORTED, RIGHTMOST, TOPMOST in ALIGN. Their pseudo-codes are shown in Algorithm 2. Given a set of objects, SORTED sorts them in increasing order of $y - x$, where (x, y) is the centroid of an object. That is, frames are sorted from the top right to left bottom by SORTED. SORTED uses both x and y coordinates, and therefore there could be frames with much smaller values for either x or y. To consider these frames, we also use RIGHTMOST and TOPMOST, which return the index of the rightmost and topmost frames, respectively.

Using these functions, ALIGN firstly classifies input frames into the following three groups: $\mathit{first\_frame}$, the first frame to be read, $\mathit{formers}$, frames to be read relatively first, and $\mathit{latters}$, the other frames to be read lastly. On horizontally flipped "Z", $\mathit{first\_frame}$ corresponds to the frame at the top right corner, $(\mathit{formers})$ to the frames in the left top area, and $(\mathit{latters})$ to the frames in the rest. Then ALIGN returns $[\mathit{first\_frame}] + \mathrm{ALIGN}(\mathit{formers}) + \mathrm{ALIGN}(\mathit{latters})$ (line 23) if the given input has more than two objects, and it operates recursively. If the input has only one object. ALIGN just returns the input itself.

Now we explain how to determine $\mathit{first\_frame}$, $\mathit{formers}$, and $\mathit{latters}$. First we identify $\mathit{first\_frame}$. Using SORTED, we sort out frames from the right top to the left bottom (line 5), and we also try to find the rightmost frame or topmost frame by TOPMOST and RIGHTMOST (line 6, 7). There are three cases: (1) If there does not exist the rightmost frame or topmost one, then we choose the first frame in the output from SORTED as $\mathit{first\_frame}$ (line 8, 9). (2) If there is either the rightmost frame or topmost one, we choose it as $\mathit{first\_frame}$ (line 10, 11). (3) If there are both the rightmost frame and topmost one, we choose the firster frame of these two frames in the order of SORTED (line 12, 13). Next, we identify $\mathit{formers}$ and $\mathit{latters}$. Based on ymax, which is the underline of an object, of $\mathit{first\_frame}$, we divide the other frames into two groups (line 16 to 22): a set $\mathit{formers}$ (resp. $\mathit{latters}$) of frames above (resp. below) that ymax.

B. *Ordering_Texts*

When we read texts of a comic, we follow the order of frames. So, we need to consider both orders of frames and texts simultaneously. To do so, we assume that "every text belongs to a frame". Based on this assumption, we define the order of texts as the sequence of texts in the first frame, then the text sequence in the second frame, . . . , then the text sequence in the last frame. In the following, we explain how to identify the corresponding frame for each text and how to sort texts in a frame.

To identify the frame to which a text belongs, we have to consider the following three patterns of the rectangle area of the text and a frame:

1) the text is totally contained in a frame,

Algorithm 1 The sort algorithm for frames in reading order

```
function ALIGN(objects)
    if $len(objects) < 1$ then
        return objects
    end if
    objects $\leftarrow$ SORTED(objects)
    most_right $\leftarrow$ RIGHTMOST(objects)
    most_top $\leftarrow$ TOPMOST(objects)
    if neither most_right nor most_top exists then
        first_frame $\leftarrow$ objects[0]
    else if both most_right and most_top exists then
        first_frame $\leftarrow$ objects [min(most_right,most_top)]
    else if either most_right or most_top exists then
        first_frame $\leftarrow$ objects [most_right or most_top]
    end if
    formers, latters = [], []
    for object in objects and object != first_frame do
        if object.ymax $\leq$ first_frame.ymax then
            formars.append(object)
        else
            latters.append(object)
        end if
    end for
    return [first_frame] + align(formers) + align(latters)
end function
```

Algorithm 2 Support functions

```
function SORTED(objects)
    for ob in objects do
        ob.score $\leftarrow$ (ob.ymax+ob.ymin) - (ob.xmax + ob.xmin)
    end for
    objects $\leftarrow$ sort(objects,key = object.score)
    return objects
end function
function RIGHTMOST(objects)
    index $\leftarrow$ None
    if others.xmax $\leq$ object.xmin then
        index $\leftarrow$ object.index
    end if
    return index
end function
function TOPMOST(objects)
    index $\leftarrow$ None
    if others.ymin $\geq$ object.ymax then
        index $\leftarrow$ object.index
    end if
    return index
end function
```

2) the text is not totally contained in any frame, but the text has overlap with some frames, or
3) the text has no overlaps.

Among these three patterns, the first case has more strong relationship between the text and the frame. Even for a frame in the third case, a text placed closer to it has more strong relationship to it than texts placed further to it. Thus, if a text is contained frames, then the closest frame from the text is selected for the text. If a text is not contained in any frame but has some overlaps with some frames, then the closest frame is selected for the text. And if a text is not contained in any frame and has no overlap with any frames, then the closest frame is selected for the text. Using this procedure, every text belongs to a frame.

Now we consider the order of texts in a frame. Similar to the order of frames in a comic, it is natural to read texts from top to bottom and right to left. So we also think that moving trajectory looks like flipped horizontally "Z", and use function ALIGN for the set of texts in a frame. Compared to frames on a page, it is noteworthy that texts are sparse. In addition, we do not need to read a text first even it is rightmost. Therefore, we do not use RIGHTMOST and TOPMOST functions in ALIGN.

C. Algorithm Evaluation

Manga109 is the main target of this paper, but it does not have reading orders of frames and texts as metadata. So we evaluate these algorithms using eBDtheque [5] that has metadata for reading orders of them. Ordering_Frames uses the coordinates of the rectangle that the frames have. In Ordering_Texts, the coordinates of the word bubbles with one or more text lines of dialog such as utterance or narration are used. From this dataset, we are available to use a subset of 6 pages met the following two criteria. The first is expression form used in comics. In the eBDtheque, there are pages without texts, and pages similar to the expression in picture books. These pages were excluded for evaluation. The second is text orientation in comics. As mentioned above, order of text differs depending on text orientation. While the pages of Manga109 are characterized by Japanese, vertical writing, and right binding, many of the pages in the eBDtheque are different. The pages that differed from these characteristics were excluded from the evaluation.

To evaluate, we use the Spearman's rank correlation coefficient between the results of our sort algorithms and orders in the eBDtheque because these orders are expressed on ordinal scale. TABLE I shows the results for each of Ordering_Frames and Ordering_Texts. In TABLE I, the correlation coefficients for the six pages and the average value for these pages are noted. Our algorithms were able to sort the frames into the correct reading order and the texts into correlation coefficients of at least 0.9. The texts in two pages were sorted incorrectly. This is because there exist some frames in which the order of some texts are back and forth.

TABLE I
EVALUATION OF OUR ALGORITHMS

	img1	img2	img3	img4	img5	img6	Ave
Frame	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Text	1.000	1.000	1.000	1.000	0.942	0.928	0.978

IV. METHODS

In this section, we explain four previous methods and three proposal methods. The proposal methods include them based on the reading order we defined. We give each method a name to make the differences easier to understand and will refer to them by those names in this paper.

A. *Previous Methods*

1) Distance_TtoC: Similar to [4], Distance_TtoC uses Euclidean distance between a text and a character. This method is based on the hypothesis that a character placed near a target text is the most plausible speaker of that text. For each text, Distance_TtoC gives a score to every character. If certain character is depicted more than once on a two-facing page, the closest character to the text is only scored. The score of a character is the reciprocal of the distance between them that is divided by 10% of the length of a diagonal of the page in Manga109. If a page has no character, no score is given.

2) Same_frame: Distance_TtoC gives a score in a concentric circle from a text. In contrast, Same_frame uses the relationship between the frames where a character and a text are placed. This method is based on the hypothesis that among characters who have the same Euclidean distance from a target text, the most plausible speaker is the character placed in the same frame as that text. This method, as in [7], scores as follows. This method assumes that a target text belongs to a frame f. If there are one or more characters in f and there are n of them, they are given a score of $\frac{1}{n}$. Note that if f has no characters, no score is given.

3) Bubble_centroid: [4] asserted that position of word bubble tail is important for the estimation. Bubble_centroid uses the tails that many word bubbles have, as shown in Fig. 2. Tails indicate the directions in which speakers are placed. In [7], the direction of a tail and its speaker are estimated as follows. First, the contour of a bubble is obtained by the method of [10]. Second, based on the curvature of the contour, the point of the bubble tail is obtained. As a result, the direction of a bubble tail is opposite the centroid of the bubble, starting from the tail. In addition, scores are given to characters who intersected the line segment with 10% of a diagonal length across the page in the direction of a tail from the tail.

4) Text_FPP&SEP: Text_FPP&SEP uses first-person pronouns and sentence-ending particles in texts that provide clues to speaker estimation. The pronouns (e.g., *Boku* and *Watashi*) and the particles (e.g., *ja* and *gozaru*) represent the characteristics of a speaker. To use them, this method requires labeled training data, unlike the three methods as above. In this study, we use part of the ground-truth dataset [6].

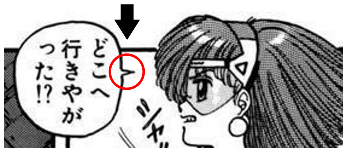

Fig. 2. The red circle indicates the tail of the word bubble. (C)ARMS/Masaki Kato [8], [9]

B. *Proposal Methods*

1) Distance_FtoF: Same_frame focuses on whether a target text is included in the same frame with a character. In contrast, Distance_FtoF foucses only the frames. This method uses not only the target frame but also those around it. Distance between frames is defined by Ordering_Frames. Distance_FtoF gives a score to every character as follows. We suppose that a target text belongs to a frame f and that it is the f_i th frame in the frame order. Let F be the set of frames $\{f_1, f_2, \cdots, f_n\}$ (order of frame $f_1 \cdots f_n \in \mathbb{N}$) in which a character places on a two-facing page. This method gives $1 \,/\, (\min(\{\mathrm{abs}(f_i - f_j) | f_j \in F\}) + 1)$ to a character, that is, the scores are given inversely proportional to the distance between the frames to which a target text or a character place in the frame order.

2) Text_2entire: While Text_FPP&SEP uses the pronouns and the particles, Text_2entire uses entire texts. Text_FPP&SEP in [7] failed to give a score to the pronouns and the particles not included in the pre-prepared corpus. To make effective use of linguistic information, we propose Text_2entire that uses entire texts. As a candidate model for this method, BERT [11], pre-trained language model with a large number of sentences, performs a task called Next Sentence Prediction in the pre-training phase. This task guesses whether two input sentences are consecutive or not. For these reasons, BERT can be described as a context-learning language model. This method inputs texts following the text order as context into BERT for speaker estimation. In Text_2entire, texts are scored as follows. If there is a text immediately preceding a target text in the text order, the two texts are used for BERT inputs; otherwise, only the target text is used. Among the output of BERT, the vector corresponding to *[cls]* is assumed to be a document vector, where *[cls]* is a classification token inserted at the first of every sequence. The score is given by a Multi-Layer Perceptron, a model of neural network, with the vector input. In this paper, BERT models are generated for each comic.

3) Bubble_contour: To estimate the direction of a bubble tail, Bubble_centroid uses centroid of a word bubble. A more hypothesis is that the directions can be determined only by the sharpest point of curvature in the contour of a bubble and its surroundings. We propose Bubble_contour that uses them. This method estimates the direction of the tail and the speakers as follows. Before allocating scores to characters, this method calculates the direction of a bubble. First, as in [10], the contour of a bubble is obtained, where the contour is represented as a point sequence $\{\boldsymbol{p}_i\}$. Next, the point $\boldsymbol{p}_t$ which is the bubble tail that is obtained based on the curvature of the contour. Finally, using point $\boldsymbol{p}_t$ and the relative vectors $\boldsymbol{q}_{t-1} = \frac{\boldsymbol{p}_{t-1}-\boldsymbol{p}_t}{||\boldsymbol{p}_{t-1}-\boldsymbol{p}_t||}, \boldsymbol{q}_{t+1} = \frac{\boldsymbol{p}_{t+1}-\boldsymbol{p}_t}{||\boldsymbol{p}_{t+1}-\boldsymbol{p}_t||}$ of the points before and after it, $\boldsymbol{v}_{\mathrm{tail}}$, the direction of a bubble tail, is obtained as follows.

$$\boldsymbol{v}_{tail} = \boldsymbol{p}_t - \frac{1}{2}\left(\boldsymbol{q}_{t-1} + \boldsymbol{q}_{t+1}\right)$$

After $\boldsymbol{v}_{tail}$ is calculated, scores are allocated in the direction of $\boldsymbol{v}_{\mathrm{tail}}$ from the tail, as Bubble_centroid.

V. Experiment

In this section, we summarize settings of experiment to evaluate the proposed methods as above.

We use the Manga109 to evaluate the methods. Since this dataset has no ground-truth data to estimate texts' speakers, we also use the dataset in [6]. We treat a two-facing page as one sample. Ordering_Frames sorts frames of a half-page at a time. Ordering_Texts uses text regions. For all estimation methods, the following data are used. Considering the methods that use bubble contours, the texts in which the contours are detected around text regions in the pages by the method of [10] are used. Of the datasets, 80% of each comic are used as training samples. In Manga109, there are characters who appear extremely infrequently throughout a comic. Similar to [7], to estimate the texts of important characters, we focus on the top five most frequently appearing characters in each comic. In addition, we excluded the data annotated as different or unknown by two annotators from the ground-truth dataset. The number of annotated texts with bubble contours detected is 147,882 of 147,918, of which 29,877 are selected 20% for evaluation. Out of them, 20,393 are the top five characters' texts, and 18,570 are annotated with the same label. Therefore, we uses these 18,570 texts to evaluate.

In this experiment, three proposed methods are compared with four previous methods. We used accuracy as a measure to evaluate the methods of speaker estimation. Note that the accuracy, to avoid weighting comics with a large amount of text, is calculated for each comic, and the average of 109 comics is taken. Each method outputs an array that sums to 1, so when combining two or more methods, simply add the values output by these methods.

VI. Results

In this section, we show the results of the proposed methods along with previous methods. Based on the results, we also evaluate the effectiveness of the proposed methods.

A. Accuracy of Each Method

First, TABLE II shows each accuracy of the previous methods. In particular, the column of *Contribution of combination* in this table mean as follows: a contribution is calculated by subtracting the accuracy of three other methods' combination from the accuracy of four methods' combination. For example, the contribution of Same_frame in this table is 0.147, which means that the three other methods' combination is 0.577 (= 0.724 - 0.147). Regarding the contribution of Bubble_centroid in this table, it is 0, but the exact value is 0.000133. This contribution demonstrated that Bubble_centroid is a valid cue for speaker estimation in the previous methods' combination.

TABLE II
Accuracy values for previous methods

	Accuracy	***Contribution of combination***
Distance_TtoC	0.614	0.052
Same_frame	0.664	0.147
Bubble_centroid	0.334	0.000[a]
Text_FPP&SEP	0.349	0.005
Conbination	0.724	

[a] Exact value is 0.000133.

TABLE III
Accuracy values for proposed methods

	Accuracy	***Contribution of combination***
Distance_FtoF	0.731	0.238
Text_2entire	0.402	-0.048
Bubble_contour	0.384	0.020
Conbination	0.692	

Second, TABLE III shows each accuracy of the proposed methods. In this table, the meaning of *Accuracy* and *Contribution of combination* is the same as in TABLE II. This table shows that the combination of three proposed methods reduce the accuracy compared to Distance_FtoF alone. In addition, the contribution of Text_2entire demonstrated that it is difficult to estimate comic speaker using only the Text_2entire.

Since Text_2entire reduce the accuracy, we focus some combinations with higher accuracy among previous and proposed methods. TABLE IV shows that some combination methods included our methods that improves the accuracy. The best accurate method in this table is a combination of Distance_FtoF, Bubble_contour, and Text_FPP&SEP.

B. Effectiveness of Proposed Methods

To evaluate effectiveness of proposed methods, we verified that each method has changed the accuracy by using a t-test. Since each method calculates the accuracy for 109 comics, a paired t-test is used and its degrees of freedom is 108. The results of t-test are shown in TABLE V.

TABLE V has two categories; first, it is mean accuracy of the two methods compared and secondly, it is the t and p values of the t-test performed under the null hypothesis that these accuracy are the same. This table shows that Distance_FtoF

TABLE IV
Accuracy values for combination of methods

	Accuracy
Distance_FtoF	0.731
Distance_FtoF + Bubble_centroid	0.732
Distance_FtoF + Bubble_contour	0.740
Distance_FtoF + Bubble_contour + Text_FPP&SEP	0.744

TABLE V
ACCURACY VALUES AND RESULTS OF T-TESTS FOR EACH METHOD

	Accuracy		***Accuracy***	t	p
Distance_FtoF	0.731	Same_frame	0.664	7.84	3.30×10^{-12}
Distance_FtoF	0.731	Distance_TtoC	0.614	15.35	6.56×10^{-29}
Distance_FtoF	0.731	Same_frame + Distance_TtoC	0.719	3.42	0.0008
Bubble_contour	0.384	Bubble_centroid	0.334	12.93	1.09×10^{-23}
Proposed combination	0.744	Previous combination	0.724	5.56	9.42×10^{-8}

significantly improved the accuracy for each of Distance_TtoC and Frame_same, as well as for the combination of the two. Bubble_contour also significantly improved it compared to Bubble_centroid. In bottom row of this table, the combination with the highest accuracy in TABLE IV are compared to the combination of methods in TABLE II. This result means that the proposed combination can be estimated more significantly than the combination of previous methods.

VII. DISCUSSION

In this study, we proposed methods based on the reading orders as a context. The accuracy of Distance_FtoF clearly showed that the Ordering_Frames contributes to the speaker estimation, and supported our hypothesis. In addition, the result of TABLE IV showed that the speaker estimation requires not only physical distance, but also frame order, word bubbles, and linguistic information. Combining multiple pieces of information from comics could help us solve this task.

In contrast, Text_2entire using Ordering_Texts is not effective, contrary to our hypothesis. Since previous study suggested that the importance of texts, we believe there is room for improvement in the evaluation of our algorithm and the input of this method. For this evaluation, we used a subset of the eBDtheque labeled the reading order because of the limited data with the same right-bound comic as manga109. Much of the data in the eBDtheque is left-bound comic and different reading order. From now on, using these data inverted right and left will be needed to strong evaluation. Besides, this method uses a target text and a previous one. Adding contextual information to the input, such as information of the previous speaker, seems to improve the estimation.

VIII. CONCLUSION

To estimate speakers in comics, we proposed algorithms for considering continuity of comics. These algorithms sort frames or texts in the reading order of a comic. An evaluation has showed that these algorithms sort frames and texts of Japanese manga with high correlation coefficients of more than 0.9.

We also proposed three methods based these algorithms for speaker estimation. Among them, a method using the order of frame especially is the highest accuracy. This result has found that a effective clue is closeness in the reading order between each frame to which a target text or a character placed. Moreover, combining information such as word bubbles and linguistic information has improved accuracy compared to only frames. Our hypothesis, the continuity of comics helps the speaker estimation, has been supported by these results.

However, it is difficult for these results to suggest that the order of texts is a clue to the speaker estimation. In future study, we plan two attempts to reconsider the algorithm and the method using text order. First, we will change the input data in this method and verify its accuracy. Second, we will expand the data for evaluation and re-evaluate this algorithm.

REFERENCES

[1] Research Institute for Publications, “Size of the comics market in 2021,” 2022/02/25, https://shuppankagaku.com/wp/wp-content/uploads/2022/02/%E3%83%8B%E3%83%A5%E3%83%BC%E3%82%B9%E3%83%AA%E3%83%AA%E3%83%BC%E3%82%B92202.pdf (Accessed Apr. 28, 2022).

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS’17. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 6000–6010.

[3] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, “Natural tts synthesis by conditioning wavenet on mel spectrogram predictions,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4779–4783.

[4] C. Rigaud, N. Le Thanh, J.-C. Burie, J.-M. Ogier, M. Iwata, E. Imazu, and K. Kise, “Speech balloon and speaker association for comics and manga understanding,” in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 351–355.

[5] C. Guérin, C. Rigaud, A. Mercier, F. Ammar-Boudjelal, K. Bertet, A. Bouju, J.-C. Burie, G. Louis, J.-M. Ogier, and A. Revel, “ebdtheque: a representative database of comics,” in *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR)*, 2013.

[6] K. Abe and S. Nakamura, “Dataset construction for mapping between texts and speakers in comics, and its challenge,” *The 3rd Meeting of Special Interest Group on Comic Computing*, pp. 7–12, 2020.

[7] K. Abe, “A research of the method to recognition the speaker of comic text,” Master’s thesis, Graduate School of Advanced Mathematical Sciences, Meiji University, 2020.

[8] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa, “Sketch-based manga retrieval using manga109 dataset,” *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 21 811–21 838, 2017.

[9] K. Aizawa, A. Fujimoto, A. Otsubo, T. Ogawa, Y. Matsui, K. Tsubota, and H. Ikuta, “Building a manga dataset “manga109” with annotations for multimedia applications,” *IEEE MultiMedia*, vol. 27, no. 2, pp. 8–18, 2020.

[10] T. Tanaka, F. Toyama, J. Miyamichi, and K. Shoji, “Detection and classification of speech balloons in comic images,” *The Journal of the Institute of Image Information and Television Engineers*, vol. 64, no. 12, pp. 1933–1939, 2010.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423