Creating Character Networks from Comics using Frames and Words in Balloons

Harumi Murakami
Graduate School of Engineering
Osaka City University
3-3-138, Sugimoto, Sumiyoshi,
Osaka 558–8585 Japan
Email: harumi@osaka-cu.ac.jp

Yusuke Nagaoka
Graduate School for Creative Cities
Osaka City University
3-3-138, Sugimoto, Sumiyoshi,
Osaka 558–8585 Japan

Ryota Kyogoku
Graduate School for Creative Cities
Osaka City University
3-3-138, Sugimoto, Sumiyoshi,
Osaka 558–8585 Japan

Abstract—We create character networks from comics using the frames and the words in balloons to find and understand them. First, we build a dataset that contains a comic's characters and its words in balloons separated by frames. Second, we extract the semantic relations among characters using the words in the balloons and the frames. Third, we develop character networks by identifying the important characters with their relatedness and add semantic relations to the networks. We implemented our approach and successfully built character networks from four popular Japanese comics and evaluated our kinship relation extraction algorithms using them.

Keywords—comics, character networks, character connections, frames, words in balloons

I. Introduction

In recent years, the popularity of Japanese manga (comics) has increased worldwide, and a large amount of it is being published. Japanese manga is usually serialized in magazines and later compiled in books. Some manga resembles long epics. Since finding a chapter, a specific volume, or even a particular manga itself from a large amount of manga is difficult, we investigate how to find a chapter or a volume of a particular manga.

Character networks, which include character charts or character connections, are often created to introduce the contents of such multimedia as movies, anime, and TV dramas to help people understand the characters and complicated stories. Character networks are also useful for finding and understanding comics. However, it remains unclear how to create character networks from comics.

This research creates character networks from comics to help users find and understand their contents.

In our previous work [1], we proposed a method of creating character networks from comics using the frequencies of characters and their co-occurrences by referring to frames. Although that pioneering work highlighted the potential of such an approach, we did not implement algorithms for creating character networks or present any for extracting the semantics among characters using texts.

In this research that uses the words included in balloons, we enhance our previous work and create character networks with semantic relations. In addition, we revise our algorithm

that creates character networks and present a new version of it.

In what follows, related work is shown in Section 2. We explain our approach's overview and our algorithms in Section 3. Experimental results using four popular Japanese comics are shown in Section 4.

II. Related Work

To the best of our knowledge, our previous work [1] with its original idea is the only research that created character networks from comics. In that work, we assumed that characters who appear in identical or in the next frames are related and created character networks using the frequencies of the characters and their co-occurrences in identical or the next frames. We used a volume of a very famous Japanese comic (Dragon Ball) for that previous experiment. In this new work, we extend our previous work by modifying our algorithm that creates character networks and adding kinship relations to our character networks. We implemented and evaluated our algorithms using four different popular Japanese comics: Naruto, Dragon Ball (different volumes from those used in the previous work), Boys Over Flowers, and Space Brothers.

Some research has created character networks from literary texts such as novels. We briefly discuss such research by selecting examples of them.

Elson et al. [2] extracted social networks from 19th century British novels and serials for literary studies. They found instances of quoted speech, attributed each quote to a character, and identified when certain characters are having a conversation. This work is related to ours in the sense that they create character networks based on the frequency of the characters and the interaction-based-weight between them. However, they did not display any semantic relations among characters.

Lee and Yeung [3] created people (characters) and place networks from literary texts. They translated the first five books of the Hebrew bible into English and used them for their evaluation. They used not only dialogue interaction but also explanatory texts as weight relations between characters. They did not display any relations between characters.

Nishihara [4] extracted the relations among characters from Japanese novels and created character networks. His work is the most closely related to ours because he deals with Japanese

and creates character networks with semantic relations such as kinship. The main difference is that he deals with novels instead of comics. Another major difference for connection generation is that he does not specify characters to display; we choose main characters with related characters to summarize a comic's content in one diagram. As for the extracting relation algorithms, he mainly extracts useful information from explanatory texts using pattern-matching. For example, the "P1 is a sister of P2" template extracts a P1-sister-P2 relation. He defined 53 relations among characters including kinship relations. Since comics generally have few explanations in their texts, we treat the texts in balloons near the characters as "dialogs" and extract kinship relations from the above information.

Kokkinakis et al. [5] identified personal relationships between main characters in 19th century Swedish fiction using information guided by named entities. They used a relationship vocabulary, which includes kinship relations, for labeling and extracted sentences with co-occurring pairs of person named entities and labeled the extracted pairs of person entities using window sizes. One way is a window size of 1-3 tokens that uses pattern matching templates with lexical units from the resources. Another way is with a window size of 4-10 tokens to measure the context similarity between the extracted pairs of person entities. Our way is different from the work of Kokkinakis et al. in a sense that we treat the texts in the balloons near the characters as dialogs, as we described in the previous paragraph.

Some research created character networks from other media. For example, Goto et al. [6] made character charts from EPG texts that introduce movies. Ogasawara et al. [7] extracted people from broadcast videos to construct a human correlation graph and examined both text and image processing. They did not implement their graphs.

Spysee2 (http://spysee2.jp/), whose original algorithm is based on Matsuo et al. [8], extracted person information from the web and displays social networks. Famous manga character names such as Naruto can be input. Users can explore the social networks created from the information about Naruto on the web, not from media contents. Even though this method is useful for well-known comics, it is inadequate to express the character networks of such designated units as chapters or volumes based on media contents.

III. METHOD

A. Overview

This research creates character networks from comics using the frames and the words in balloons to find and understand them. First, we built a dataset that contains a comic's characters and its words in balloons separated by frames. Second, we extracted the semantic relations among characters using the words in the balloons and the frames. Third, we developed character networks by identifying the important characters by their relatedness and added semantic relations to the networks.

The main feature of this research is creating character networks from comics using the frequencies of characters and their co-occurrences by referring to the frames and identifying the relations between characters using the words in balloons.
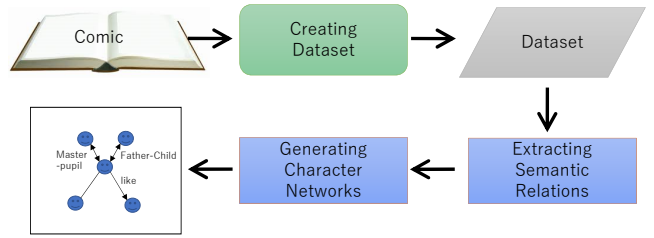

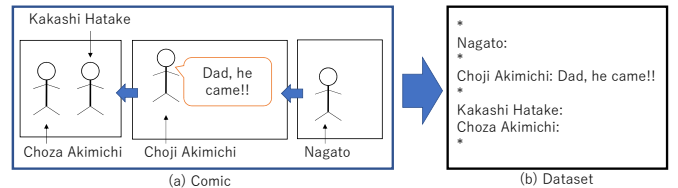Fig. 1. Overview of our approach.

Fig. 2. Example of creating a dataset.

No current work creates character networks from comics except for our previous work.

An overview of our approach is shown in Figure 1. In the following, we illustrate our approach using the famous comic Naruto Vols. 46-50. Note that the examples presented in this paper were translated into English for publication.

B. Creating Dataset

We manually create a comic dataset comprised of sets of a character and his/her words in balloons that are divided by frames.

For example, as in Figure 2, a dataset is created from a comic. Note that Japanese comics flow from top to bottom and right to left.

C. Extracting Semantic Relations among Characters

We extract the semantic relations among characters using a heuristic method based on matching words and dictionaries.

As an initial step, we focus on kinship by capturing calls (such as father or mother). For example, if the words in balloons contain the word father and satisfy certain conditions, the speaker is a child. We assume that the recipient of this call will be characters in the same or in the next frames.

1) Extracting Relations using Kinship Dictionary: We illustrate our method with the following child-father relation.

Step 1. Matching whether a term in the dictionary (e.g., dad, father) appears in the words.

Table I shows our dictionary with six family relations. Note that in Japan, although younger siblings use brother or sister to denote their older brother/sisters, older siblings seldom use these terms for younger brother/sisters.

Step 2. When a term exists in words, if the previous term is not *no* (means "of") and the following term is not a particle, we assume that the speaker is talking to his father.

TABLE I.   Kinship dictionary

Relation (abbreviation)	Sample terms	From	To
child-father (C-F)	dad, father	child	father
child-mother (C-M)	mom, mother	child	mother
person-older brother (P-OB)	brother	person	older brother
person-older sister (P-OS)	sister	person	older sister
grandchild-grandfather (GC-GF)	grandpa	grandchild	grandfather
grandchild-grandfather (GC-GM)	grandma	grandchild	grandmother

TABLE II.   Extracted relations using kinship dictionary

No.	Character 1	Character 2	Relation	Score
1	Choji Akimichi	Nagato	child-father	3
2	Choji Akimichi	Kakashi Hatake	child-father	2
3	Choji Akimichi	Choza Akimichi	child-father	4
...	...	...	...	...
5	Shima	Fukasaku	child-father	4
...	...	...	...	...
13	Kiba Inuzuka	Akamaru	child-mother	2
14	Kiba Inuzuka	Tsume Inuzuka	child-mother	2
...	...	...	...	..
17	Fukasaku	Shima	child-mother	4
...	...	...	...	...
19	Nagato	Nagato's mother	child-mother	1

Step 2.1 When there are other characters in the same frame as the speaker, a point is given to the relations between the speaker and the other characters in the frame.

Step 2.2 When there is only one character (speaker) in the frame, a point is given to the relations between the speaker and other characters in the next (before or after) frames.

In the example of Figure 2, since Dad is included in the child-father dictionary and condition Step 2 is also satisfied, relations are created between Choji Akimichi and Nagato, Kakashi Hatake, and Choza Akimichi. In this case, Choji Akimichi who said "Dad" becomes a child and Nagato, Kakashi Hatake, and Choza Akimichi become fathers.

Table II shows a part of the extracted relations from Naruto Vols. 46-50. Twelve child-father and seven child-mother relations were extracted.

2) Identifying Spouse and Modifying Relations: In Japan, married couples with children often call each other "father" or "mother" as if they are nicknames. We also use this custom for our work. If two characters call each other "father" and "mother," they are identified as a married couple, a new married couple relation (in this paper, a wife-husband relation) is created and the previous child-father and child-mother relations are deleted. In other words, if character 1 and character 2 have child-father relation points and character 2 and character 1 have child-mother relation points, character 1 is a wife and character 2 is a husband.

In the example in Table II, a wife-husband relation between Shima and Fukasaku is created. The original two Shima between Fukasaku (child-father and child-mother) relations are deleted.

3) Modifying Child-Parent Relations: In general, a person has just one father and one mother. We select the most likely child-parent relations based on the candidate relations.

If a person has multiple child-father or child-mother relations, the relations are selected with maximum scores. If multiple candidates remain, we perform pattern-matching on the character names and select the relation that has the most similar name. If the similarity is identical, all of the candidates are deleted.

Now in our examples, we selected Choji Akimichi and Choza Akimichi as the child-father since its score is maximum (4), and other two child-father candidate relations are deleted.

There are four child-mother candidate relations whose child is Kiba Inuzuka. First, Kiba Inuzuka-Katsuyu is deleted because its score is 1 (not maximum). Next, name similarity chooses Tsume Kinuzuka as Kiba Inuzuka's mother.

We obtained the result relations shown in No. 1-6 in Table III.

4) Modifying Relations using Thresholds: Since the main and related characters appear many times, many (relatively) unimportant relation candidates might be extracted. We set thresholds to delete such relations.

First, we classified the characters into five groups by their cumulative appearance rates. In our Naruto example, there are a total of 8,089 character appearances. Naruto Uzumaki (the main character) appears 1,210 times, and his appearance rate is 0.15 (1,210/8,089). Nagato appears 1,200 times, and his cumulative appearance rate is (1,210+1,200)/8,089 = 0.30. These two characters are classified in group 1 (see Table IV).

Next we set thresholds for each group using Eq. (1). For our Naruto example, *Num* = 4/2 = 2. The threshold for group 1 is 2, for group 2 it is 1, and for groups 3 to 5, it is 0.

$$Num = \frac{max(score)}{2}.$$

$$Th_{group} = Num - (group - 1).$$

Here $group$ is a group number (i.e., 1, 2, 3, 4 or 5).

If Th_{group} is under 0, then set zero to Th_{group}:

$$Th_{group} = \begin{cases} \text{round up to pos integer} & (Th_{group} > 0) \\ 0 & (Th_{group} \leq 0) \end{cases}. \quad (1)$$

TABLE III.   Results of extracted relations

No.	Character 1	Character 2	Relation	Score	Judge
1	Choji Akimichi	Choza Akimichi	child-father	4	✓
2	Kakashi Hatake	Sakumo Hatake	child-father	3	✓
3	Naruto Uzumaki	Minato Namikaze	child-father	3	✓
4	Kiba Inuzuka	Tsume Inuzuka	child-mother	2	✓
5	Fukasaku	Naruto Uzumaki	child-mother	2	× error
6	Shima	Fukasaku	wife-husband	–	✓
7	Fukasaku	Minato Namikaze	grandchild-grandfather	–	× error (with inference only)

Precision: 5/6 = 0.833 (without inference); 5/7 = 0.714 (with inference)

TABLE IV.   RANKING OF CHARACTER APPEARANCES

Rank	Character	Appearance	Group	Display
1	Naruto Uzumaki	1,210	1	✓
2	Nagato	1,200	1	✓
3	Sakura Haruno	264	2	✓
4	Sasuke Uchiha	253	2	
5	Kakashi Hatake	251	2	✓
5	A	251	2	
7	Tobi	202	3	
8	Sai	172	3	✓
9	Onoki	167	3	
10	Danzo	154	3	
11	Fukasaku	152	3	✓
12	Konan	141	3	✓
13	Killer Bee	140	3	
...	...	...	...	
26	Yahiko	88	4	✓
27	Jugo	86	4	
28	Jiraiya	84	4	✓
28	Yamato	84	4	
30	Shizune	83	4	
31	Minato Namikaze	81	4	✓
32	Shikamaru Nara	76	4	✓
...	...	...	...	

TABLE V.   INFERENCE RULES

A father of a father is a grandfather.
A mother of a mother is a grandmother.
A father of a mother is a grandfather.
A mother of a mother is a grandmother.
People who share a father are brothers or sisters.
People who share a mother are brothers or sisters.
A brother/sister of a father is an uncle/aunt.
A brother/sister of a mother is an uncle/aunt.
Brothers or sisters have identical parents.

Third, we set thresholds for each relation using the minimal character threshold:

$$Th_{character} = Th_{group}$$

$$Th_{relation_{AB}} = min(Th_{character_A}, Th_{character_B}).$$

For example, for the Naruto Uzumaki-Minato Namikaze relation, the threshold for Naruto Uzumaki is 2 and for Minato Namikaze is 0, and the threshold for their relation becomes 0. Since the score of this relation (3) exceeds the threshold (0), it is not deleted.

In our example, since the scores of all the examples exceed their threshold, no relation is deleted in Table III.

D. Adding Relations using Inference

Family inference is a typical example for artificial intelligence. Such relations as grandparent-child can be defined using basic parent-child or spouse relations. Table V shows the inference rules used in this research.

With the above rules, person-uncle and person-aunt relations can be generated without the kinship dictionary.

In our Naruto example, from the Naruto Uzumaki-Minato Namikaze child-father relation and the Fukasaku-Naruto Uzumaki child-mother relation, a Fukasaku-Minato Namikaze grandchild-grandfather relation can be created.

Table III shows the final result for our Naruto example. We extracted six relations without inference and added one relation by the inference process. Five of six relations (without inference) are correct and out of seven (with inference) are correct.

E. Creating Character Networks

Various networks can be drawn, and we present an overview network that summarizes the contents of the comics in a single diagram.

We assume that critical characters appear more frequently than other characters. We select the three most frequent characters first and then display three related characters for them. Finally, lines and semantic relations are drawn among characters. Creating a character network algorithm is described below.

Step 1. Three "main" characters whose appearance is in the top three are displayed up, down-left, and down-right.

Step 2. Three other related characters whose relatedness (Eq. (2), see below) is in the top three for the three main characters (except for the main characters) are selected. These other characters are displayed next to the three main characters. When they are related to more than one main character, they are displayed between the related main characters.

Step 3. Lines are connected from the main characters to the related displayed characters (including main and related characters).

Step 4. The extracted relation names (e.g., child-father) are described next to the relations.

Calculating the relatedness among characters, which was originally presented in our previous work [1], is redefined as follows.

The relatedness score between characters c_i and c_j is calculated as the summation of their related scores using the co-appearance inside the frames and the scores using the appearance in the next frames, as shown in Eq. (2).

$$R_{inside}(c_i, c_j) = \begin{cases} 1/n & (n = 1, 2, 3, 4) \\ 0.25 & (n \geq 5) \end{cases}$$

where n is the number of different characters from the designated character inside a frame.

$$R_{next}(c_i, c_j) = \begin{cases} 0.5/n & (n = 1, 2, 3, 4) \\ 0.25 & (n \geq 5) \end{cases}$$

where n is the number of different characters from the designated character in the next frame.

$$R(c_i, c_j) = \sum_{k=1}^{n} R_{inside}(c_i, c_j) + \sum_{k=1}^{n-1} R_{next}(c_i, c_j). \quad (2)$$

Figure 3 shows the character network that is automatically created from our Naruto example. Naruto Uzumaki (main

Rank	Character 1	Character 2	Score	Display
1	Naruto Uzumaki	Nagato	442.292	✓
2	Naruto Uzumaki	Sakura Haruno	98.658	✓
3	Nagato	Konan	91.400	✓
4	Naruto Uzumaki	Fukasaku	91.342	✓
5	Naruto Uzumaki	Minato Namikaze	85.683	✓
6	Killer Bee	Kisame Hoshigaki	83.917	
6	Naruto Uzumaki	Kakashi Hatake	66.625	✓
8	Nagato	Yasuke	62.858	✓
9	Nagato	Jiraiya	58.767	✓
10	Naruto Uzumaki	Katsuyu	55.292	
11	Naruto Uzumaki	Sai	52.567	
12	Sasuke Uchiha	A	51.267	
13	Naruto Uzumaki	Karui	51.217	
14	Karui	Omoi	47.308	
15	Sakura Haruno	Sai	43.358	✓
16	Sasuke Uchiha	Karin	37.183	
17	Nagato	Kakashi Hatake	35.433	
...	...	...	...	
52	Kakashi Hatake	Sakura Haruno	17.550	✓
...	...	...	...	
62	Sakura Haruno	Shikamaru Nara	14.558	✓
...	...	...	...	

Character 1	Character 2	Relation	Judge
Choji Akimichi	Choza Akimichi	child-father	✓
Kakashi Hatake	Sakumo Hatake	child-father	✓
Naruto Uzumaki	Minato Namikaze	child-father	✓
Kiba Inuzuka	Tsume Inuzuka	child-mother	✓
Shima	Fukasaku	wife-husband	✓
Nagato	Nagato's father	child-father	× not extracted
Nagato	Nagato's mother	child-mother	× not extracted
Ino Yamanaka	Inoichi Yamanaka	child-father	× not extracted

Recall: 5/8 = 0.625 (without and with inference)

IV. Experiment

We successfully built character networks from four Japanese comics. Figure 3 shows the output from the Naruto dataset. In this section, we focus on the evaluation of our algorithms for creating semantic relations.

A. Method

We built four datasets from four comics: Naruto, Vols. 46-50, Dragon Ball (kanzenban) Vols. 29-34, Boys Over Flowers, Vols. 16-20, and Space Brothers, Vols. 1-5. The average number of characters (four characters in Figure 2) was 10,985. 6,581 characters (one character in Figure 2) out of 10,985 had words in balloons (60.1%).

We evaluated our algorithms with and without inferences.

The following are the evaluation measures for extracting relations:

$$Precision = \frac{\text{correct extracted relations}}{\text{extracted relations}}$$

$$Recall = \frac{\text{correct extracted relations}}{\text{correct relations to be extracted}}$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall}.$$

For our Naruto example, Table III shows how to calculate Precision and Table VII shows how to calculate Recall.

We generated six kinship relations, five of which were correct without inference, so the precision without inference is 0.833. Seven (one more) relations were generated with inference, five of which were correct, so the precision with inference is 0.714 (Table III). Eight kinship relations exist in the dataset, and five were extracted, so the recall is 0.625 (Table VII).

B. Results and Analysis

In all four comics, we identified the following relations and their amounts: 19 child-father, 18 child-mother, 11 wife-husband, 3 person-older brother, 2 person-older sister, 2 grandchild-grandfather, 1 grandchild-grandmother, and 1 person-uncle. We identified no person-aunt relations.

Table VIII shows the experiment results without/with inference. Overall, without inference is better in precision and f-measure and with inference is better in recall.
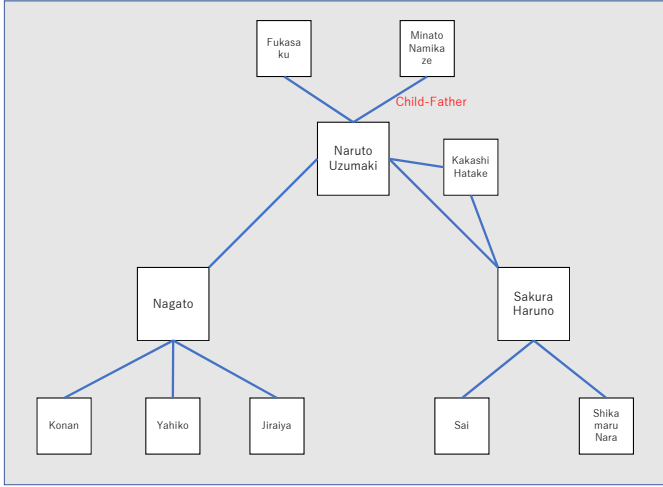

Fig. 3.   Generated overview character networks for Naruto Vols. 46-50 (without inference).

character), Nagato, and Sakura Haruno are displayed as the three main characters. Since Kakashi Hatake is selected as being among the top three with Naruto Uzumaki and Sakura Haruno, he is displayed between them. The child-father relation name is displayed alongside the line between Naruto Uzumaki and Minato Namikaze. Four infrequent characters, as categorized in group 4 (Yahiko, Jiraiya, Minato Namikaze, and Shikamaru Nara), are displayed in the character network from their relatedness to the main characters.

Table IV lists the ranking of character appearances, and Table VI ranks the relatedness between characters. Please check these tables for the characters and relations (marked as "Display" in the final column) that appear in Figure 3.

Note that we chose without inference for automatic display networks based on our experiment.

TABLE VIII. EXPERIMENT RESULTS

Comic	Without inference			With inference		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Naruto	0.833	0.625	0.714	0.714	0.625	0.666
Dragon Ball	0.666	0.166	0.265	0.714	0.208	0.322
Boys Over Flowers	0.368	0.583	0.451	0.303	0.833	0.444
Space Brothers	0.411	0.411	0.411	0.269	0.411	0.325
Average	0.569	0.446	0.460	0.500	0.519	0.439

TABLE IX. RESULTS OF THREE RELATIONS

Relation	Without Inference			With Inference		
	Precision	Recall	F-measure	Precision	Recall	F-measure
child-father	0.688	0.500	0.579	0.500	0.591	0.542
child-mother	0.714	0.333	0.454	0.412	0.389	0.400
wife-husband	0.500	0.364	0.421	0.500	0.364	0.421

Table IX shows the results for three relations. For the child-father and child-mother relations, without inference is better in precision and f-measure and with inference is better in recall. For wife-husband relations, nothing was added in the inference process, and the result was identical.

We explain the following three types of errors using our Naruto example: precision, recall, and inference.

For the precision error, Fukasaku and Naruto Uzumaki appear together many times. Since Fukasaku calls his wife Shima "mother," Naruto Uzumaki-Fukasaku (C-M) was extracted.

For the recall error, we failed to extract Nagato-Nagato's father (C-F), Nagato-Nagato's mother (C-M), or Yamanaka-Inoichi Yamanaka (C-F). There was only one "father call" and one "mother call" from Nagato to other characters. In these cases, Nagato's father and mother were in the same frame. Our algorithms could not identify which person was the father or the mother. For the Yamanaka case, there was no "father call" in the dataset.

From the Naruto Uzumaki-Minato Namikaze child-father relation and the Fukasaku-Naruto Uzumaki child-mother relation, a Fukasaku-Minato Namikaze grandchild-grandfather relation was created. Basically, inference error occurs from incorrect relations. One apparent solution is that the user should check and correct his/her answers before the inference process and do based on correct relations.

From the precision and recall failure analysis, we identified room for improvement and the following possible solutions: (1) add terms to the dictionary or modify the call extraction heuristics to properly capture the calls; (2) modify how to score the relations; (3) extract relations without using calls.

V. DISCUSSION

Finally, we discuss the contributions of our research.

First, to the best of our knowledge, our work (including our previous work) is the only research that creates character networks from comics. We modified our previous algorithm for creating character networks and added kinship relations to them.

Second, we implemented our algorithms and successfully built overview character networks from four popular Japanese comics: Naruto, Dragon Ball (using different volumes from our previous work), Boys Over Flowers, and Space Brothers.

Third, we evaluated our algorithm for extracting kinship relations and found that without inference is better in precision and f-measure and with inference is better in recall. Without inference is suitable for automatic generation, and inference should be used after correcting the initial relations.

Even though basic algorithms using frames are applicable to any language, some heuristics of extracting semantic relations depend on culture and language. For example, for wife-husband relation extraction, we used Japanese married couples with children who refer to each other as "father" and "mother." We must consider heuristics for other cultures and languages.

Future work will improve the extraction of relation algorithms. We need to evaluate how to generate overview networks algorithms and consider other relations, such as "friends" or "rivals," to make character networks more useful. Other types of character networks such as "group" should also be developed in the future.

VI. CONCLUSIONS

We presented a method to create character networks from comics using the frames and the words in balloons to identify and understand comics. First, we built a dataset that contains a comic's characters and their words in balloons separated by frames. Second, we extracted the semantic relations among characters using the words in balloons and frames. Third, we developed character networks by identifying critical characters by their relatedness and added semantic relations to the networks. We implemented our approach and successfully built character networks from four popular Japanese comics. We also evaluated kinship relation extraction algorithms using these comics.

REFERENCES

[1] H. Murakami, R. Kyogoku, and H. Ueda, "Creating character connections from manga," in *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence (ICAART 2011)*, vol. 1, 2011, pp. 677–680.

[2] D. K. Elson, N. Dames, and K. R. McKeown, "Extracting social networks from literary fiction," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, 2010, pp. 138–147.

[3] J. Lee and C. Y. Yeung, "Extracting networks of people and places from literary texts," in *Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation (PACLIC 2012)*, 2012, pp. 209–218.

[4] H. Nishihara, "Extraction of relations among characters in literary texts," Master's thesis, JAIST, 2015.

[5] D. Kokkinakis and M. Malm, "Character profiling in 19th century fiction," in *Proceedings of Language Technologies for Digital Humanities and Cultural Heritage Workshop*, 2011, pp. 70–77.

[6] J. Goto, Y. Yagi, A. Aizawa, and S. Sekine, "Generation of correlation charts from tv programs based on anaphora resolution," in *The 22nd Annual Conference of the Japanese Society for Artificial Intelligence*, 2008.

[7] T. Ogasawara, T. Takahas, I. Ide, and H. Murase, "Construction of a human correlation graph from broadcasted video," in *The 22nd Annual Conference of the Japanese Society for Artificial Intelligence*, 2005.

[8] Y. Matsuo, M. Mori, J.and Hamasaki, K. Ishida, T. Nishimura, H. Takeda, K. Hashida, and M. Ishiduka, "Polyphonet: An advanced social network extraction system," in *Proceedings WWW2006*, 2006, pp. 397–406.