

Creating Character Networks with Kinship Relations from Comics

Harumi Murakami ^{*}, Yusuke Nagaoka ^{*}, Ryota Kyogoku ^{*}

Abstract

To find and understand comic contents, we present an approach of creating character networks with semantic relations from comics using the panels and speech balloons. First, we build a dataset that contains a comic's characters and its words in speech balloons separated by the panels. Second, we extract the semantic relations among characters using the words in the speech balloons and panels. Third, we develop character networks by identifying the important characters using the frequencies of characters and their relatedness, and then we add semantic relations to these networks. As an initial step in finding semantic relations, we focus on kinship relations. We implement our approach and successfully build character networks from four popular Japanese comics. Finally, we evaluate our kinship relation extraction algorithms and the created character networks.

Keywords: manga, comics, character networks, panels, speech balloons

1 Introduction

The popularity of manga (Japanese comics) has increased worldwide [1]. Many manga have been translated into other languages, and a large number of manga are being published. For example, over 1 million copies of “Pokémon: The Electric Tale of Pikachu” issue #1 have been sold in the United States, making it the best-selling comic book there since 1993 [2]. Manga are usually serialized in magazines and later compiled in books. Some manga resemble long epics. For example, the popular manga “Naruto” was serialized in the magazine Weekly Shōnen Jump from 1993 vol. 43 [3] to 2014 vol. 50 [4], later to be compiled into 72 books. A total of 700 episodes appear in the 72 books. Since finding an episode, a volume (book) of a manga, or even a particular manga itself from a large-scale manga collection is difficult, we investigate ways of finding a particular episode or volume (book) of a particular manga, as well as a manga series itself.

Character networks, which include character correlation charts or diagrams, are often created to introduce the contents of TV dramas in Japan to viewers, thus helping them better understand the characters and stories. Character networks for comics, movies, animation, novels, etc., can also be found on the web. These are created manually for the same purpose

^{*} Osaka City University, Osaka, Japan

as the TV networks. We believe that such character networks are useful for finding and understanding the contents of comics. However, it is unclear how to most effectively create character networks from comics. On the other hand, there exists much research on creating character networks from literary texts such as novels. It is obvious that character names and semantic relations between characters can be extracted from quotes or explanatory text using natural language understanding techniques. Since comics are not originally constructed from texts but from images in which text information is attached, it remains unclear how to identify character names from images and how to extract dialogues from speech balloons; furthermore, there tends to be little explanatory text aside from that in the speech balloons.

This research creates character networks from comics to help users (1) find an episode, a volume of comics, or a comic itself and (2) understand the contents of episodes or volumes of comics, or the comics themselves. In this research we use the term comics instead of manga because our research is also applicable to western comics.

In our previous work [5], we proposed a method of creating character networks using the frequencies of characters and their co-occurrences by referring to panels¹. Although that pioneering work highlighted the potential of such an approach, we did not implement algorithms for drawing character networks or present any for extracting the semantics among characters using texts. In this research that uses the words included in speech balloons, we enhance our previous work and create character networks with semantic relations. In addition, we revise our algorithm that lays out character networks and present a drawing algorithm that creates a network called an “overview character network” to display important characters and overview the contents in a single diagram.

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. We discuss the significance of our research in Section 5.

2 Related Work

The research on creating character networks can be categorized by the target, such as comics, texts, and videos. Furthermore, the process of creating character networks can be divided into two steps: (1) extracting relations (including semantic relations) between characters and (2) drawing character networks. In this section, we start our discussion by focusing on these two steps for each target.

2.1 Creating Character Networks from Comics

To the best of our knowledge, our previous work [5] with its original idea is the first research that created character networks from comics. In that work, we manually created a dataset that includes characters divided by panels. We assumed that characters who appear in the same or in the adjacent panels are related and calculated the relatedness between characters using co-occurrences in the same or adjacent panels. Finally, we presented a layout of character networks using the frequencies of characters and the relatedness between characters. We used a volume of a famous Japanese comic (Dragon Ball vol. 32 [6]) in the experiment.

In this paper, we extend our previous work by (1) using words in speech balloons to extract semantic relations between characters and (2) presenting new algorithms that

¹We used the term “frame” in the previous work but we changed it to “panel” since panel is more precise for describing our work.

create character networks with semantic relations. We implemented and evaluated our new algorithms using four different popular Japanese comics: Naruto, Dragon Ball (different volumes from that used in the previous work), Boys Over Flowers, and Space Brothers. This paper is based on our conference papers [7] [8]. Detailed definitions of algorithms, explanations, and further discussion have been added. In addition, new character networks are presented.

After our work, [9] is the only research we know of that has created character networks from comics. They presented a method of creating character networks based on the speech timing of characters. They manually extracted pairs of a character name and its speech timing from three Japanese comics (one volume for each comic). They calculated relatedness between two characters based on similarity of speech timing. Their networks did not link characters directly but instead linked a character and a “group,” which is a concept of grouped related characters. They did not use any panel information or text and did not extract any semantic relations.

Since no other work has created character networks from comics, we discuss character networks created from other media.

2.2 Creating Character Networks from Literary Texts

Much research has created character networks from literary texts such as novels. We briefly discuss such research by selecting examples of them. The main differences between research on comics and that on literary texts are (1) research on literary texts extracts character names and relations (including semantic relations) automatically, and (2) it extracts information from quoted text and/or body text.

Elson *et al.* [10] 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.

Nishihara [11] extracted the relations among characters from Japanese novels and created character networks. His work is the most closely related to ours among the research efforts on texts 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 within their text, we treat the text in speech balloons near the characters as dialogues and extract kinship relations from the above information.

Kokkinakis *et al.* [12] 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 speech balloons near the characters as dialogs, as we described in the previous paragraph. In their later work, Kokkinakis *et al.* [13] created a simplified co-occurrence network for the characters in the Swedish novel “Vi Bookar, Krokår och Rothar.” They did not add any semantic relations in the network.

Lee and Yeung [14] 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. Their networks can be distinguished from others in the sense that they drew place nodes.

2.3 Creating Character Networks from Other Media

Some research created character networks not from the contents themselves but from external texts. This text-based approach is applicable to multimedia contents including comics and videos.

Goto *et al.* [15] created human correlation charts from TV program abstracts for movies in Electronic Program Guides (EPG). Although the targets (dramas and movies) are multimedia, this research uses text as well as natural language understanding techniques. The main feature of this study is that anaphora resolution is used to identify (merge) characters. Their character network uses groups of words for semantic relations, not only nouns such as “only daughter” but also verbs such as “loves,” “be married,” and “declare an affair.”

Ramakrishna *et al.* [16] built character networks to analyze differences with respect to the characters’ gender, race, age, and other metadata in movies. They obtained movie scripts from the Internet along with additional metadata such as cast, genre, writers, and directors; moreover, they collected actor-level demographic information such as gender, race and age. Their demo of character network visualization is available on the web (https://sail.usc.edu/~mica/project_pages/character_network_visualization.html). The shape and color of character objects vary according to gender and race. Important characters are computed by network analysis, and links between characters are based on dialogues.

Spysee2 (<http://spysee2.jp/>), whose original algorithm was based on Matsuo *et al.* [17], extracted person information from the web and then displayed social networks. Famous manga character names such as Naruto can be input. Users can explore the social networks created from the information about Naruto available 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 within such designated units as episodes or volumes based on media contents.

Although it is more challenging than using external texts, some research created character networks from videos, and we describe two examples of such research as follows.

Park *et al.* [18] built a network structure called Character-Net from Video. It finds characters in a group of shots, extracts the speaker and listeners in the scene, represents it with character-based graphs and draws the relationship between all characters by accumulating the character-based graphs at video. The characters are identified by face recognition. The names of the characters can be detected using matching technology between subtitles and script. The major characters are identified by network analysis. They used three videos for their experiment.

Yuan *et al.* [19] presented a graph-based correlation search framework to analyze actor association, which enables semantic-level user navigation to better understand sitcom contents. They developed a new retrieval mode, character correlations search, that is deployed in their VisualCar actor-concurrence search system. They performed face detection and tracking on cluster faces. The clustered faces are manually adjusted for actor indexing. They proposed a context-based correlation analysis in shot sequence and a hierarchical concurrence measurement for refinement. They used the “Friends” (20 hours) sitcom in their experiments.

2.4 Computational Research on Comics

We have discussed how to create character networks from comics and other media. As the remaining body of related work, we briefly give an overview of the computational research on comics. Matsushita [20] called research on digital comics “comic computing,” and in another work [21] selected 53 papers for classifying comic computing into three issues: (1) comic coding, (2) utilizing obtained knowledge, and (3) utilizing comic presentation. Our previous work [5] was classified into a subcategory of (1): understanding the structure of comic contents.

Using the current technology, it is impossible to extract character names with words in speech balloons separated by panel information. In our research, we manually identified panels and speakers (characters) of speech balloons and created a dataset. Ueno and Kazama [9] also manually identified speech timing. If we consider the entire required technology as separate elements such as extracting panels or words in speech balloons, there exists some previous research. For example, Pang *et al.* [22] proposed a robust method for automatically extracting panels from digital manga pages. Guo *et al.* [23] extracted text strings from comic strips. Chu and Li [24] detected faces in manga. We can use these technologies as preprocessing to create a dataset. However, the difficult parts include identifying characters from images (some characters are drawn from the back view, without a face image), giving names to identified characters, and identifying speakers and listeners in relation to speech balloons. At least in the immediate future, manual effort is inevitable in creating a useful dataset. One possible solution is combining comic images with manual annotation. Fujimoto *et al.* [25] created Manga109, a dataset covering a variety of Japanese comic books, totaling 109, publicly available for academic purposes. This dataset provides numerous comic images but lacks annotations, so they have presented web-based software for efficient annotations.

3 Method

3.1 Overview

This research creates character networks with semantic relations from comics using the panels and speech balloons to find and understand comic contents. First, we build a dataset that contains a comic’s characters and its words in speech balloons separated by panels. Second, we extract the semantic relations among characters using the words in the speech balloons and the panels. Third, we develop character networks by identifying the important characters using the frequencies of characters and their relatedness, and then we add semantic relations to the networks. An overview of our approach is shown in Figure 1. As an initial step in finding semantic relations, we focus on kinship relations such as father-child

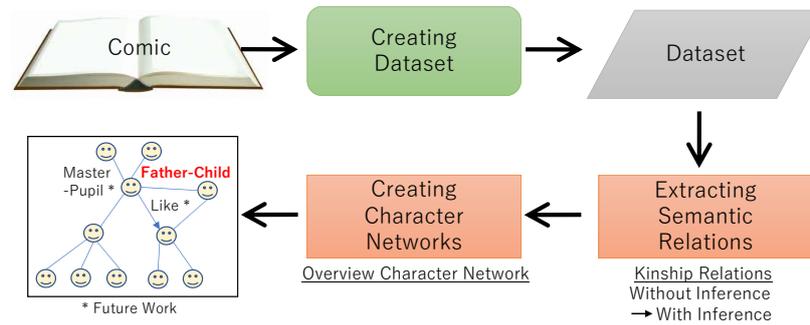


Figure 1: Overview of our approach

relations. The process of extracting semantic relations is divided into two stages: one without inference and the other with inference. As a first example of character networks, we present a network called an overview character network that displays important characters to overview the comic’s contents in a single diagram. The main feature of this research is creating character networks from comics: (a) identifying the semantic relations between characters using the words in speech balloons and panels and (b) identifying the relatedness between characters using their co-occurrences in the panels.

In the following, we illustrate our approach using the popular comic *Naruto* vols. 46 [26] - 50 [27]. Note that the examples presented in this paper were translated into English for publication.

3.2 Creating Dataset

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

Figure 2 shows an example of creating a dataset. Note that Japanese comics flow from top to bottom and right to left. We identify panels and characters inside panels with words in speech balloons and write them down in a dataset. In the dataset, the sequence of panels is divided by “*”. A character with a speech balloon is written as “*character name: words in speech balloons*”. A character without a speech balloon is written as “*character name:*”. Character names that cannot be identified are assigned as A, B, C, . . . , for counting the number of characters used in the algorithms.

3.3 Extracting Semantic Relations

We aim to extract semantic relations among characters. As an initial step, we focus on kinship relations. In this research, we try to capture calls (such as “father” or “mother”) using the words in speech balloons and panels. For example, if the words in speech balloons contain the word father and satisfy certain conditions, the speaker must be a child. We assume that the recipient of this call will be characters in the same or in the next panels. If there is one or more characters in the same panel with the speaker, they are considered candidates of the call’s recipient and we give a point to them. If there is no character in the same panel, we give a point to the characters in the adjacent (previous and next) panels. In our preliminary experiment, we tried various patterns of giving a point to characters,

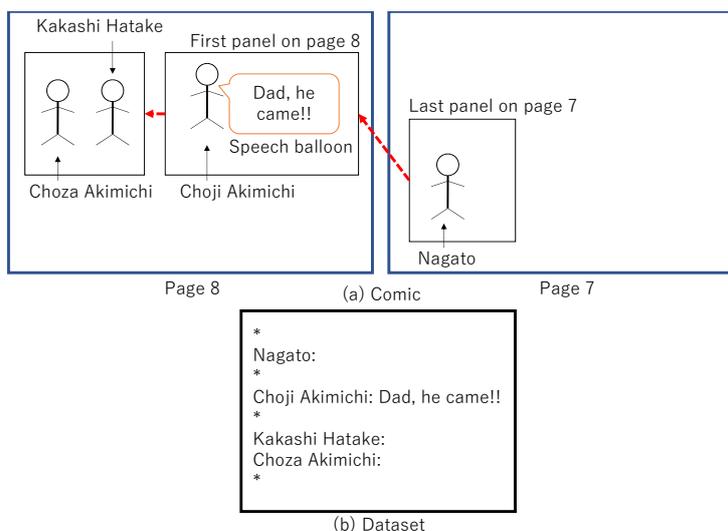


Figure 2: Example of creating a dataset

such as in the same panel only, previous panel only, next panel only, the same and adjacent panels, etc. Consequently, we chose the proposed algorithm because it provided the best performance.

3.3.1 Extracting Semantic 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 1 shows our dictionary with six family relations. Since there were few grandchild-grandparent relations in the dataset and the words *ojiichan* (grandpa) and *obaachan* (grandma) in Japanese are often used to call to elderly people (not actual grandparents), we just added neutral terms for grandfather and grandmother. For similar reasons, we did not create a dictionary for person-uncle or person-aunt relations, since the words *ojisan* (uncle) and *obsan* (aunt) in Japanese are often used to call to adults (not actual uncles or aunts). We expected to extract these four relations with inference processes. Likewise, words such as *oniisan* (brother) or *oneesan* (sister) are used to call to young adults. However, we decided to add many terms to build person-older brother and person-older sister dictionaries, since siblings are important in the dataset in our experiment. Note 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.

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

Step 2.2 When there is only one character (speaker) in the panel, a point is given to the semantic relations between the speaker and other characters in the adjacent (previous and next) panels.

In the example of Figure 2, since Dad is included in the child-father dictionary and condition Step 2 is also satisfied, semantic relations are created between Choji Akimichi

Table 1: Kinship dictionary

Relation	From	To	No. of terms	Sample terms
child-father	child	father	13	dad, father
child-mother	child	mother	13	mom, mother
person-older brother	person	older brother	17	brother
person-older sister	person	older sister	17	sister
grandchild-grandfather	grandchild	grandfather	1	grandfather
grandchild-grandmother	grandchild	grandmother	11	grandmother

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.

Figure 3 shows how to calculate initial scores for child-father relations between (from - to) characters. Here, (a) corresponds to the real example in Figure 2. The speaker of the speech balloon is Choji Akimichi. Since there is no other character with Choji Akimichi in the second panel, 1 is added to characters in the first and third panels. On the other hand, (b) is an imaginary example used to explain what happens if Choza Akimichi exists in the second panel. In this case, 1 is added to only this character in the same panel with the speaker Choji Akimichi.

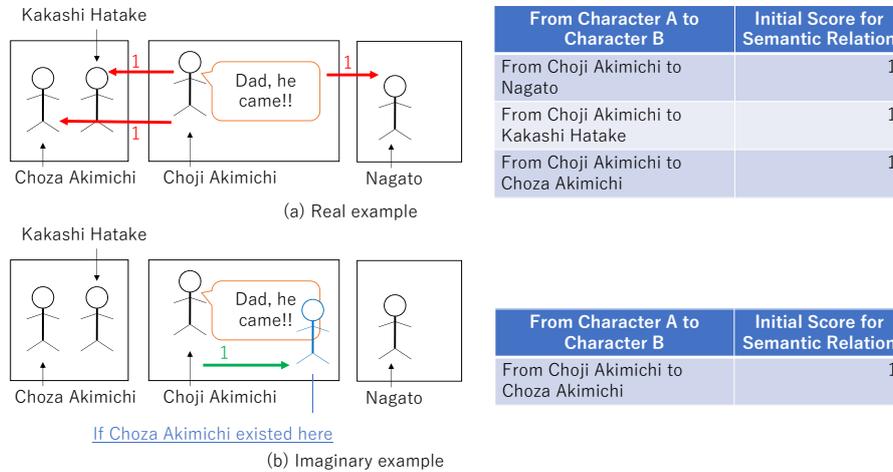


Figure 3: Example of calculating initial semantic relations

Table 2 shows the extracted semantic relations from Naruto vols. 46-50. Twelve child-father and eight child-mother relations were extracted.

3.3.2 Identifying Spouse and Modifying Semantic 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

Table 2: Extracted semantic 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
4	Kakashi Hatake	Sakumo Hatake	child-father	3
5	Shima	Fukasaku	child-father	4
6	Shima	Nagato	child-father	3
7	Shima	Naruto Uzumaki	child-father	3
8	Naruto Uzumaki	Minato Namikaze	child-father	3
9	Nagato	Nagato's father	child-father	1
10	Nagato	Nagato's mother	child-father	1
11	Onoki	Danzo	child-father	1
12	Onoki	Gaara	child-father	1
13	Kiba Inuzuka	Akamaru	child-mother	2
14	Kiba Inuzuka	Tsume Inuzuka	child-mother	2
15	Kiba Inuzuka	Kuromaru	child-mother	2
16	Kiba Inuzuka	Katsuyu	child-mother	1
17	Fukasaku	Naruto Uzumaki	child-mother	2
18	Fukasaku	Shima	child-mother	4
19	Nagato	Nagato's father	child-mother	1
20	Nagato	Nagato's mother	child-mother	1

(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 must be a wife and character 2 must be a husband.

In the example in Table 2, a wife-husband relation between Shima and Fukasaku is created. The original two Shima between Fukasaku (child-father and child-mother) relations are deleted.

3.3.3 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 relations.

If a person has multiple child-father or child-mother relations, the relations are selected with maximum scores. If multiple relations 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 relations are deleted.

From three potential relations when the child is Choji Akimichi in Table 2, we selected Choji Akimichi and Choza Akimichi as the child-father relation, since this relation's score is the maximum (4) while the other two child-father relations were deleted.

There are two child-father relations in which the child is Shima. Since their scores are identical, the name similarity for each relation is calculated. The similarity is also identical

(0), so the two relations were deleted. Likewise, we deleted two child-father relations for the child Nagato, two child-father relations for the child Onoki, and two child-mother relations for the child Nagato.

There are four child-mother 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 semantic relations shown in Table 3.

Table 3: Extracted semantic relations after section 3.3.3

No.	Character 1	Character 2	Relation	Score
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
6	Shima	Fukasaku	wife-husband	–

3.3.4 Modifying Semantic Relations using Thresholds

Since the main and related characters appear many times, many (relatively) unimportant relations might be extracted. We set thresholds to delete such relations. We want to define a function $T_{c,2}(c_i, c_j)$ that obtains a threshold between two characters c_i and c_j .

First, we classified the characters into five groups: 1, 2, 3, 4, and 5 for the top 20%, 40%, 60%, 80%, and other, respectively, according to their cumulative appearance rates. When a character's rate lies in multiple groups, the character is classified into the upper group. 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 15% (1,210/8,089). Nagato appears 1,200 times, and his cumulative appearance rate is between 15% and 30% ((1,210 + 1,200)/8,089). These two characters are classified in group 1 (see Table 4).

Next we set thresholds for each group.

$$T(g_k) = \frac{\max(\text{score})}{2} - (k - 1),$$

where g is a group and k is the group number (i.e., 1, 2, 3, 4 or 5).

If $T(g_k)$ is under 0, then set zero to $T_{group}(g_k)$:

$$T_{group}(g_k) = \begin{cases} \text{round up to pos integer} & \text{if } T(g_k) > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

For our Naruto example, $\max(\text{score})$ for extracted relations is 4 (see Table 2). The threshold for group 1 is 2, for group 2 it is 1, and for groups 3 to 5 it is 0 from Eq. (1).

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

$$T_{c,1}(c_i) := T_{group}(g_k),$$

where c_i is a character included in the group g_k (see examples of $T_{c,1}(c_i)$ in Table 4).

Table 4: Ranking of character appearances

Rank	Character	Appearance	Group	Threshold	Display
1	Naruto Uzumaki	1,210	1	2	✓
2	Nagato	1,200	1	2	✓
3	Sakura Haruno	264	2	1	✓
4	Sasuke Uchiha	253	2	1	
5	Kakashi Hatake	251	2	1	✓
5	A	251	2	1	
7	Tobi	202	3	0	
8	Sai	172	3	0	✓
9	Onoki	167	3	0	
10	Danzo	154	3	0	
11	Fukasaku	152	3	0	✓
12	Konan	141	3	0	✓
13	Killer Bee	140	3	0	
...
26	Yahiko	88	4	0	✓
27	Jugo	86	4	0	
28	Jiraiya	84	4	0	✓
28	Yamato	84	4	0	
30	Shizune	83	4	0	
31	Minato Namikaze	81	4	0	✓
32	Shikamaru Nara	76	4	0	✓
...

Finally, we set the threshold between two characters c_i and c_j to minimal thresholds for c_i and c_j as shown in Eq. (2):

$$T_{c,2}(c_i, c_j) = \min(T_{c,1}(c_i), T_{c,1}(c_j)). \quad (2)$$

For example, for the Naruto Uzumaki-Minato Namikaze child-father 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 3.

3.3.5 Adding Semantic Relations using Inference

Family inference is a typical example for artificial intelligence. Such relations as grandparent-grandchild can be defined using basic parent-child or spouse relations. Table 5 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

Table 5: 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.

grandchild-grandfather relation can be created.

Table 6: Extracted final semantic relations and precision judgment

No.	Character 1	Character 2	Relation	Judge
1	Choji Akimichi	Choza Akimichi	child-father	✓
2	Kakashi Hatake	Sakumo Hatake	child-father	✓
3	Naruto Uzumaki	Minato Namikaze	child-father	✓
4	Kiba Inuzuka	Tsume Inuzuka	child-mother	✓
5	Fukasaku	Naruto Uzumaki	child-mother	× 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 6 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.

3.4 Creating Character Networks

Various networks can be drawn, and we present an overview character network that displays important characters in a single diagram. This new algorithm is different from the algorithm presented in our previous work [5].

We assume that critical characters appear more frequently than other characters. We select the three most frequent characters (we call them “main” characters²) first and then display three related characters for them. Finally, lines and semantic relations are drawn among characters. Creating an overview 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.

²We distinguish main characters and “main” characters in this paper. The main characters are the original meaning and the “main” characters are the three characters displayed at center in the overview character networks.

Step 2. Three other related characters whose relatedness (Eq. (3), see below) is in the top three for the “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 other characters.

Step 4. Lines are connected between the “main” characters whose relatedness are over the threshold³.

Step 5. The relation names are described next to the relations. We use Father-Child for child-father relations, Mother-Child for child-mother relations, Married Couple for wife-husband relations, Sibling for person-older brother, person-older sister, brother or sister relations, Grandfather-Grandchild for grandchild-grandfather relations, Grandmother-Grandchild for grandchild-grandmother relations, Uncle-Nephew/Niece for person-uncle relations, and Aunt-Nephew/Niece for person-aunt relations.

Calculating the relatedness among characters, which was originally presented in our previous work [5], is redefined as follows. Here, we want to define a function $R(c_i, c_j)$ that calculates a relatedness score between characters c_i and c_j . 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 panels and the scores using the appearance in the adjacent panels, as shown in Eq. (3). The basic ideas are that (1) characters who exist in the same panel are very related, (2) characters who exist in an adjacent panel are somewhat related, and (3) the relatedness decreases according to the number of characters in the same or in an adjacent panel.

First, a function $R_{inside}(c_i, c_j)$ defines the inside scores:

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

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

Next, a function $R_{adjacent}(c_i, c_j)$ defines the adjacent⁴ scores:

$$R_{adjacent}(c_i, c_j) = \begin{cases} 0.5/n & \text{if } n = 1, 2, 3, 4, \\ 0.1 & \text{if } n \geq 5, \end{cases}$$

where n is the number of different characters from the designated character c_i in the adjacent panel.

Finally, $R(c_i, c_j)$ defines the related scores:

$$R(c_i, c_j) = \sum_{k=1}^m R_{inside}^{(k)}(c_i, c_j) + \sum_{k=1}^{m-1} R_{adjacent}^{(k,k+1)}(c_i, c_j), \quad (3)$$

where the summation occurs between the designated panels. For example, if three panels are given, R_{inside} is done three times (first, second, and third panels), and $R_{adjacent}$ is done twice (between first and second panels and between second and third panels).

³This threshold is not the same as that described in Section 3. In this research, we set 3 to the threshold for drawing by checking the relatedness data. We need to evaluate this in the future.

⁴We used the term “next” in the previous work but we changed it to “adjacent” since adjacent is more precise for describing our work.

Figure 4 shows how to calculate relatedness scores between characters using the same examples in Figure 3. In the real example (a), the relatedness between Kakashi Hatake and Choza Akimichi is 1 ($1 + 0$), since these two are in the same panel once. In the imaginary example, the relatedness score between Kakashi Hatake and Choza Akimichi (described as Choza Akimichi and Kakashi Hatake in order of appearance in Figure 4, which is identical since there is no direction of relatedness) is 1.5: 1 (between two characters in the third panel) + 0.5 (between Choza Akimichi in the second panel and Kakashi Hatake in the third panel).

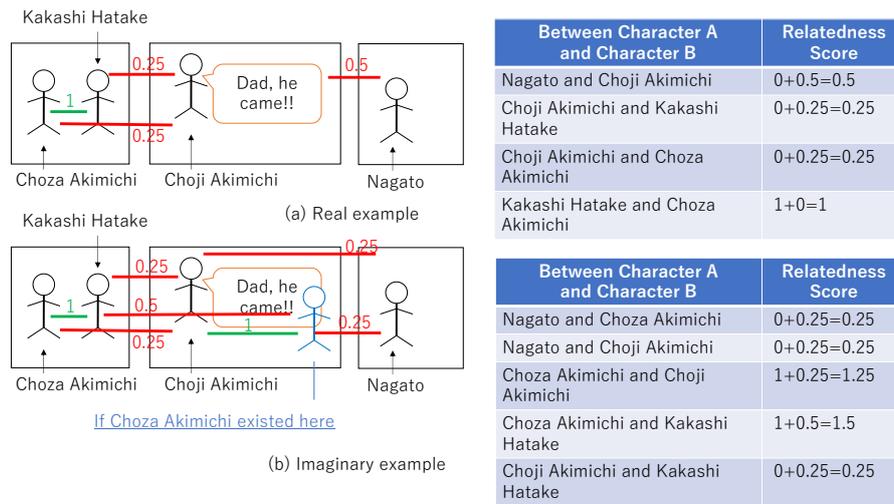


Figure 4: Exmple of calculating relatedness

After removing incorrect semantic relations from the extracted results manually, an overview character network is created for Naruto (Figure 5⁵). Naruto Uzumaki (main 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. No line is drawn between Nagato and Sakura Haruno because their relatedness is below the threshold. Father-Child 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 4 lists the ranking of character appearances, and Table 7 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 5.

4 Experiment

We successfully built overview character networks from four Japanese comics. In this section, we first focus on the evaluation of algorithms for extracting semantic (kinship) relations. Next, we analyze the generated overview character networks.

⁵In this paper’s figures, a space inside names is replaced by an underbar for implementation reasons.

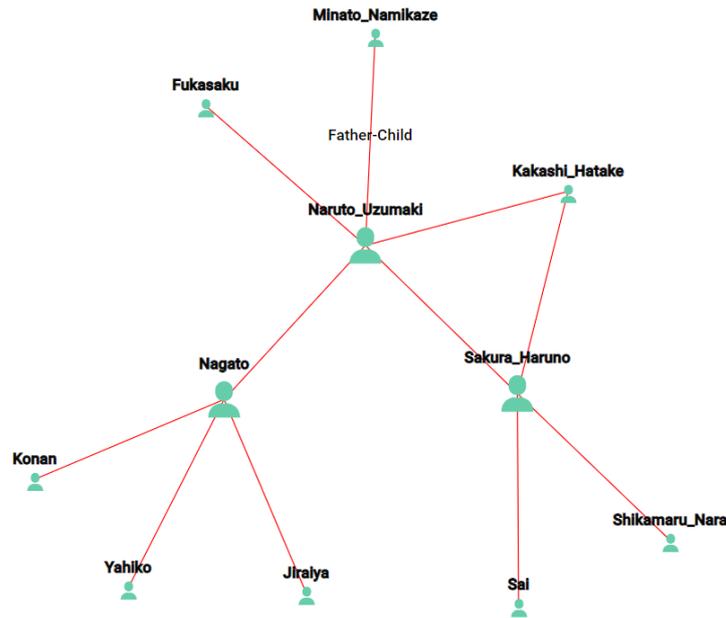


Figure 5: Overview character network created for Naruto vols. 46-50

4.1 Method

We built four datasets from four comics: Naruto, vols. 46-50, Dragon Ball (kanzenban) vols. 29 [28] - 34 [29], Boys Over Flowers, vols. 16 [30] - 20 [31], and Space Brothers, vols. 1 [32] - 5 [33]. The reasons for selecting these comics are as follows: (1) they contain a certain amount of kinship relations, (2) they cover various comic types by readers: boys (Naruto, Dragon Ball), girls (Boys Over Flowers), and youth (Space Brothers), and (3) they cover various sections of comics: opening (Space Brothers), middle (Naruto, Boys Over Flowers), and closing (Dragon Ball). The average number of total characters (four characters in Figure 2) was 10,985. Furthermore, 6,581 characters (one character in Figure 2) out of these 10,985 had speech balloons and were identified as speakers (60%).

In all four comics, We identified the following semantic relations and their amounts: 22 child-father, 18 child-mother, 11 wife-husband, 4 person-older brother, 2 person-older sister, 2 grandchild-grandfather, 1 grandchild-grandmother, and 1 person-uncle. We identified no person-aunt relations.

We evaluated our algorithms without and with inferences. The following are the evaluation measures for extracting semantic relations:

$$Precision = \frac{\text{number of correct extracted relations}}{\text{number of extracted relations}},$$

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

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}.$$

For our Naruto example, Table 6 shows how to calculate Precision and Table 8 shows how to calculate Recall. We generated six kinship relations, five of which were correct

Table 7: Ranking of character relatedness

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	✓
...

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 6). Eight kinship relations exist in the dataset, and five were extracted, so the recall is 0.625 (Table 8).

4.2 Results and Analysis

4.2.1 Extracting Semantic Relations Algorithms

Table 9 shows the experimental results for semantic (kinship) relations without and with inference for each comic. On average, precision is 0.500 (with) to 0.570 (without), recall is 0.447 (without) to 0.520 (with), and F_1 is 0.440 (with) to 0.461 (without). Overall, without inference is better in precision and F_1 and with inference is better in recall. Dragon Ball is an exception, where with inference is better in all measures. Among the four comics, the results for Naruto are the best. We believe this is because the number of relations is smallest (8) and all relations (child-father, child-mother, and wife-husband) are relatively easy to extract. On the other hand, the results for Dragon Ball are the worst. We believe this is because the number of relations is the largest (24) and various types of semantic relations, such as grandchild-grandfather, grandchild-grandmother, person-older brother, and person-uncle, are included. In addition, we observed that there exist fewer speech

Table 8: Correct semantic relations to be extracted and recall judgment

No.	Character 1	Character 2	Relation	Judge
1	Choji Akimichi	Choza Akimichi	child-father	✓
2	Kakashi Hatake	Sakumo Hatake	child-father	✓
3	Naruto Uzumaki	Minato Namikaze	child-father	✓
4	Kiba Inuzuka	Tsume Inuzuka	child-mother	✓
5	Shima	Fukasaku	wife-husband	✓
6	Nagato	Nagato's father	child-father	× not extracted
7	Nagato	Nagato's mother	child-mother	× not extracted
8	Ino Yamanaka	Inoichi Yamanaka	child-father	× not extracted

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

balloons than in other comics and that the characters tend to call each other by name rather than by dictionary terms such as father or mother. We assume that these features affected the results. Furthermore, we used the closing part of Dragon Ball to create its dataset. We assume that there is less information on semantic relations in the closing part than in the other parts of comics, since such information has already been explained in the first half of the story.

From the inference process, 25 relations were created. Unfortunately, 22 relations were incorrect and only three relations were created correctly: Son Goten-Son Goku child-father relation (Dragon Ball), Tsukushi Makino-Haruo Makino child-father relation (Boys Over Flowers), and Susumu Makino-Chieko Makino child-mother relation (Boys Over Flowers). In addition, Tsukasa Domyouji-Tsubaki Domyouji sibling relation was created (Boys Over Flowers). In Table 9, we counted as correct a person-older sister relation in convenience. This is because sibling relation and person-older brother/sister relations are redundant and sibling relation is excluded from the evaluation.

Table 10 shows the results for each semantic relation. Since there exist few results except for child-father, child-mother, wife-husband, person-older brother, and person-older sister relations, we analyzed just these five relations. For the child-father and child-mother relations, without inference is better in precision and F_1 while with inference is better in recall. For wife-husband relations, nothing was added in the inference process, and the results were identical. Person-older brother and person-older sister relations proved much worse than child-father and child-mother relations in precision. We believe this is because (1) words such as brother and sister are more frequently included than father or mother in NOT call context and (2) there are few selection processes for relations other than threshold compared to child-father or child-mother relations.

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

The precision error occurs when there are two characters within the same or adjacent panels and kinship terms (terms in a kinship dictionary) are included in a speech balloon. Under this condition, the errors can be classified into the following types: (a) When the extraction (call) is not correct (e.g. calling a man using brother, calling oneself using father), a wrong relation is created. (b) When the recipient of a call is incorrect, an incorrect relation is created. (c) When a wrong relation is chosen among the relation candidates. (d) When there exist one or more children with a father and/or a mother call, a wrong wife-husband

Table 9: Results of semantic relations for each comic

Comic	Without inference			With inference		
	Prec	Rec	F ₁	Prec	Rec	F ₁
Naruto	0.833 (5/ 6)	0.625 (5/ 8)	0.714	0.714 (5/ 7)	0.625 (5/ 8)	0.667
Dragon Ball	0.667 (4/ 6)	0.167 (4/24)	0.267	0.714 (5/ 7)	0.208 (5/24)	0.323
Boys Over Flowers	0.368 (7/19)	0.583 (7/12)	0.452	0.303 (10/33)	0.833 (10/12)	0.444
Space Brothers	0.412 (7/17)	0.412 (7/17)	0.412	0.269 (7/26)	0.412 (7/17)	0.326
Average	0.570	0.447	0.461	0.500	0.520	0.440

Table 10: Results of semantic relations for each semantic relation

Relation	Without Inference			With Inference		
	Prec	Rec	F ₁	Prec	Rec	F ₁
child-father	0.688 (11/16)	0.500 (11/22)	0.579	0.500 (13/26)	0.591 (13/22)	0.542
child-mother	0.750 (6/ 8)	0.333 (6/18)	0.462	0.412 (7/17)	0.389 (7/18)	0.400
wife-husband	0.500 (4/ 8)	0.364 (4/11)	0.421	0.500 (4/ 8)	0.364 (4/11)	0.421
person-older brother	0.182 (2/11)	0.500 (2/ 4)	0.267	0.182 (2/11)	0.500 (2/ 4)	0.267
person-older sister	0.200 (1/ 5)	0.500 (1/ 2)	0.286	0.333 (2/ 6)	1.000 (2/ 2)	0.500
grandchild-grandfather	0.000 (0/ 0)	0.000 (0/ 2)	0.000	0.000 (0/ 3)	0.000 (0/ 2)	0.000
grandchild-grandmother	0.000 (0/ 0)	0.000 (0/ 1)	0.000	0.000 (0/ 1)	0.000 (0/ 1)	0.000
person-uncle	0.000 (0/ 0)	0.000 (0/ 1)	0.000	0.000 (0/ 1)	0.000 (0/ 1)	0.000
person-aunt	0.000 (0/ 0)	0.000 (0/ 0)	0.000	0.000 (0/ 0)	0.000 (0/ 0)	0.000

Table 11: Results for semantic relations in overview character networks for each comic

Comic	Full-automatic			Semi-automatic		
	Prec	Rec	F ₁	Prec	Rec	F ₁
Naruto	0.500 (1/ 2)	1.000 (1/ 1)	0.667	1.000 (1/ 1)	1.000 (1/ 1)	1.000
Dragon Ball	1.000 (2/ 2)	1.000 (2/ 2)	1.000	1.000 (2/ 2)	1.000 (2/ 2)	1.000
Boys Over Flowers	0.250 (1/ 4)	1.000 (1/ 1)	0.400	1.000 (1/ 1)	1.000 (1/ 1)	1.000
Space Brothers	0.500 (1/ 2)	0.500 (1/ 2)	0.500	1.000 (1/ 1)	0.500 (1/ 2)	0.667
Average	0.563	0.875	0.642	1.000	0.875	0.917

Full automatic contains without/with inference results.

relation is created. In Naruto, Fukasaku and Naruto Uzumaki appear together many times in the same panels. Fukasaku calls his wife Shima “mother,” but more points were given to Naruto Uzumaki, and thus the Fukasaku-Naruto Uzumaki child-mother relation was extracted. This is the type (d) error.

The recall errors can be classified into four types: (a) When two characters do not appear within the same or adjacent panels. (b) When there is no speech balloon between two characters. (c) When the extraction (call) fails (e.g. kinship terms are not included). (d) When a wrong relation is chosen or nothing is chosen among relations. We failed to extract Nagato-Nagato’s father child-father, Nagato-Nagato’s mother child-mother, and Ino Yamanaka-Inoichi Yamanaka child-father relations. Moreover, 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 panel, but our algorithms could not identify which person was the father or the mother. These are type (d) errors. For the Yamanaka case, there was no “father call” in the dataset. This error is type (c). We did not find any case where the “When the recipient of a call is wrong” error (type (b) precision error). We believe that for our algorithm our basic idea, i.e. “giving an initial point to characters in the same panel and when there is no other character in the same panel giving it to characters in the adjacent panels,” is reasonable.

All inference errors occurred from incorrect relations. As an example in Naruto, 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. One apparent solution would be for people to check and correct their answers before the inference process and then make inferences based on correct relations.

From the precision and recall failure analysis, we identified room for improvement and the following possible solutions: (a) add terms to the dictionary (for grandchild-grandparent, person-uncle/aunt relations, in particular); (b) modify the initial extraction algorithms to properly capture the calls; (c) modify the algorithms of selecting relations; (d) extract relations using explanatory text. We assume that (b) and (d) will greatly improve the results by using more careful natural language understanding processing. For example regarding (b), we can add such a condition as “when a kinship term is followed by *tachi* (plural), no relation should be created.” For example regarding (d), from such a speech balloon as “Oh,

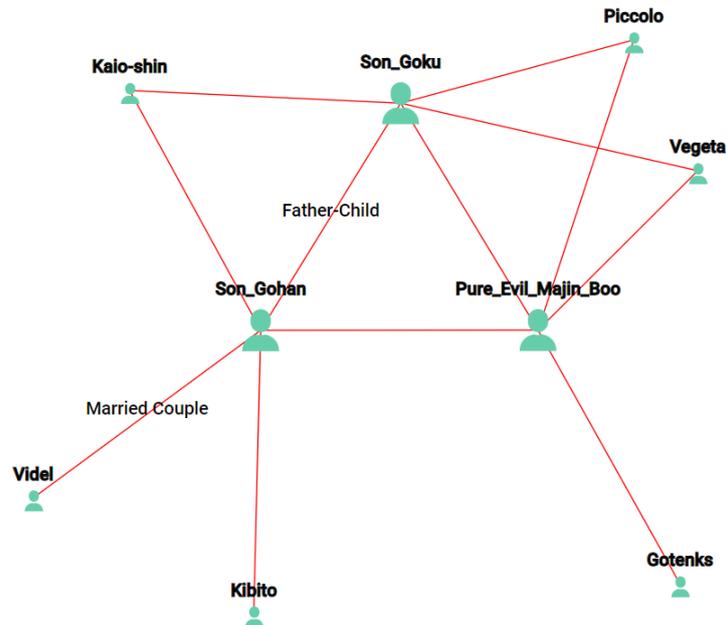


Figure 6: Overview character network created for Dragon Ball vols. 29-34

yes! Hibito is my younger brother!” the Hibito Nanba-Mutta Nanba person-older brother relation can be extracted.

4.2.2 Created Character Networks

We attempted (a) full-automatic display (without inference), (b) full-automatic display (with inference), and (c) semi-automatic display (display after removing wrong semantic relations). Table 11 shows the results on narrowing down the target to the semantic relations to be drawn in the overview character networks. The full-automatic results contain both without and with inference cases; in other words, the results are the same in full-automatic displays (a) and (b). Precision is 0.563, recall is 0.875, and F_1 is 0.642 in full-automatic displays, and precision is 1.000 (naturally), recall is 0.875, and F_1 is 0.917 in semi-automatic display. Since limited semantic relations are drawn in overview character networks, the results are much higher than those in Table 9.

When we used full-automatic display (without/with inference), five wrong labels were displayed: Fukasaku-Naruto Uzumaki child-mother (Naruto), Shigeru Okawahara-Shigeru’s mother child-father, Akira Mimasaka-Tsukasa Domyouji person-older brother, Tsukushi Makino-Rui Hanazawa person-older brother (Boys Over Flowers), and Reiji Nitta-Yasushi Furuya person-older brother (Space Brothers) relations. One recall error was not extracting the Hibito Nanba-Mutta Nanba person-older brother relation (Space Brothers). Dragon Ball showed perfect results. When we used semi-automatic display, Naruto and Boys Over Flowers attained perfect results as well as Dragon Ball.

We analyze the four networks in detail by referring to Figures 5, 6, 7, and 8, which were created by (c) semi-automatic display.

Naruto belongs to the genre of battle manga. Naruto is the main character. Naruto vols. 46-50 tell the story of the fight between Naruto and Nagato. Naruto and Nagato are

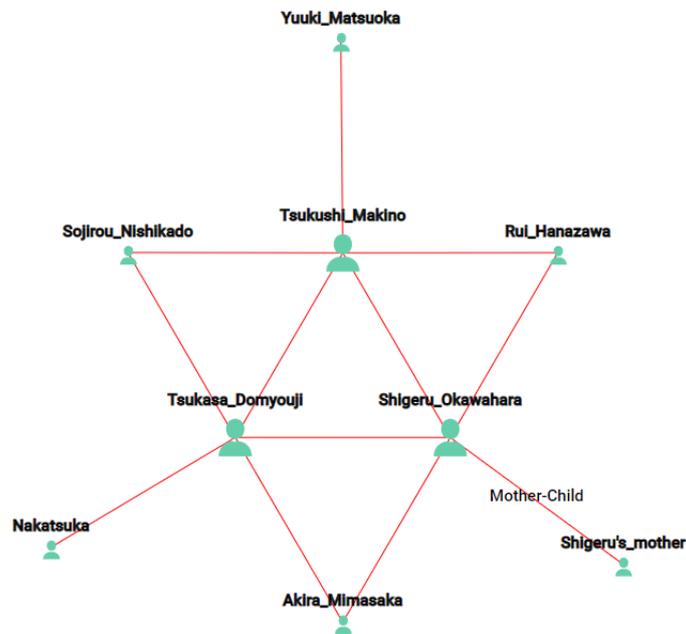


Figure 7: Overview character network created for Boys Over Flowers vols. 16-20

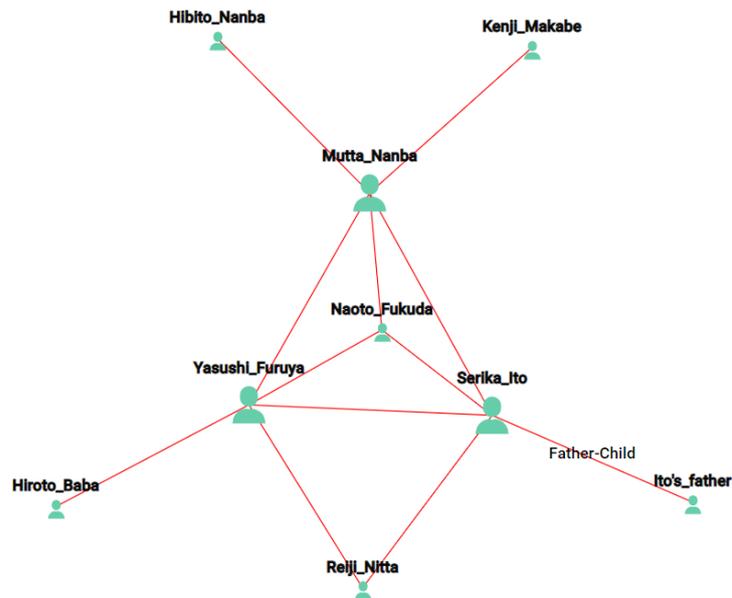


Figure 8: Overview character network created for Space Brothers vols. 1-5

displayed as “main” characters. Their teacher is the same, Jiraiya. Nagato, Konan, and Yahiko form a group of isolated enemies who fight against Naruto (Figure 5). Nagato and Sakura Haruno are almost always in different places and thus they are not linked together.

Dragon Ball can also be classified as battle manga. Son Goku is the main character. Dragon Ball (kanzenban) vols. 29-34 tell the story of Majin Boo. In Figure 6, Son Goku and Pure Evil Majin Boo, who is one of the different forms of Majin Boo, are displayed as “main” characters. In this figure, Pure Evil Majin Boo is the only enemy, and the other characters except for Videl and Kibito fought him.

Boys Over Flowers is a romantic comedy manga. The main character is Tsukushi Makino. The second main character is Tsukasa Domyouji, a boy who loves her. Rui Hanazawa, Sojiro Nishikado, and Akira Mimasaka are third-level main characters, and they and Tsukasa Domyouji are called “F4 (flower-four).” Boys Over Flowers vols. 16-20 involve the story of a triangular relationship among Tsukushi Makino, Tsukasa Domyouji, and Shigeru Okawahara (Tsukasa’s fiancée). Figure 7 shows the love triangle of Tsukushi Makino, Tsukasa Domyouji and Shigeru Okawahara as “main” characters. Nakatsuka is an enemy in this network while the other characters are good friends. Here, F4 members are also displayed.

Space Brothers is a narrative manga. Mutta Nanba and Hibito Nanba are brothers, as the first and second main characters, who aim to become astronauts. Space Brothers vols. 1-5 tell the story of Mutta Nanba challenging himself toward becoming an astronaut along with other candidates. In Figure 8, the “main” characters Mutta Nanba, Yasushi Furuya, and Serika Ito are on the same team in the astronaut program. Naoto Fukuda is also on the same team and linked to all three “main” characters. Reiji Nitta is on the same team too.

5 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 with semantic relations from comics. The main feature of this research is creating character networks from comics using the words in the speech balloons and panels. Second, we evaluated our algorithm for extracting kinship relations and found that without inference is better in precision and F_1 and with inference is better in recall. Third, 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.

Even though basic algorithms using panels 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.

Here, we describe future work. First, we need to improve the extraction of kinship relation algorithms. We have already suggested ideas for improvement in Section 4.2.1. Second, we need to evaluate overview character networks in more detail. Third, we need to consider other semantic relations, such as “friends” and “rivals,” to make character networks more useful. As Figure 1 shows, “likes” might be a one-way relation, and thus the type of relations should be considered in the future.

We have already presented overview character networks, but we could also use our

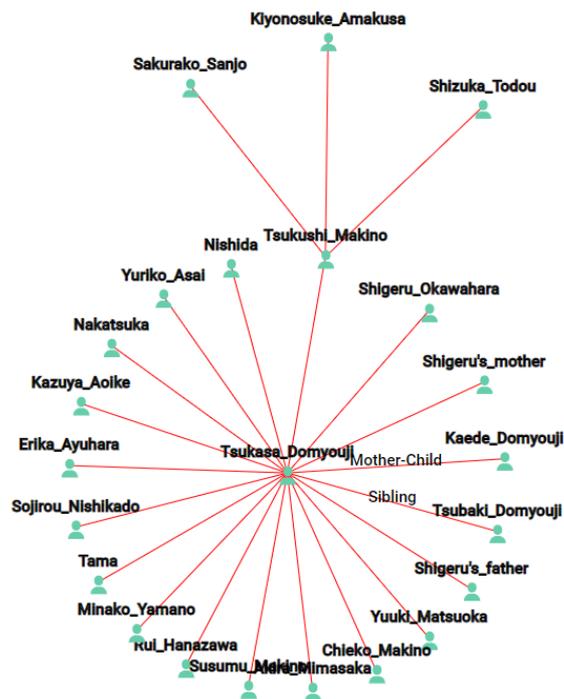


Figure 9: Single-focus character network created for Tsukasa Domyouji in Boys Over Flowers, vols. 16-20

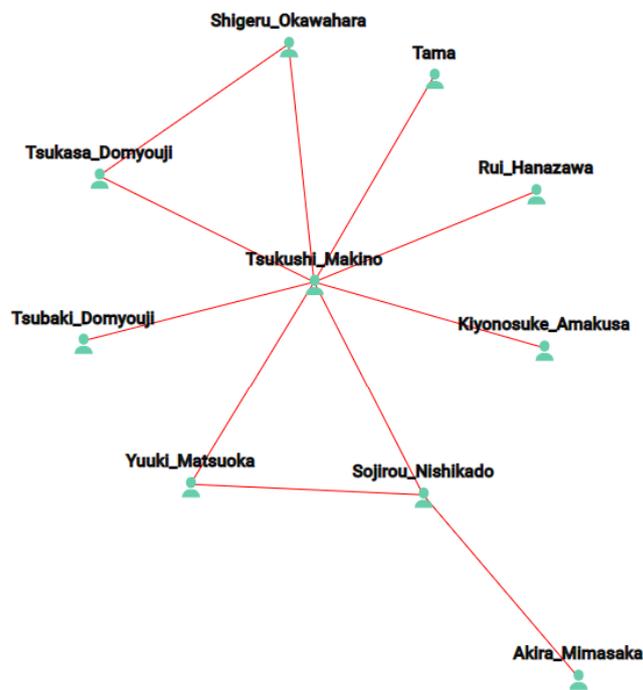


Figure 10: Frequent co-occurrence character network created for Boys Over Flowers, vols. 16-20

approach to create other possible types of networks: (a) single-focus character networks and (b) frequent co-occurrence character networks. Next, we explain our semi-automatic examples using *Boys Over Flowers* to show a different example than *Naruto*.

If users wanted to focus on one character, a network centering on that one character would be useful. Figure 9 shows such an example of a “single-focus character network.” Kaede Domyouji (mother) and Tsubaki Domyouji (sister) are linked from Tsukasa Domyouji (second main character in *Boys Over Flowers*). Our algorithm allows users to set link depths, and breadth-first search is applied to each link depth using relatedness. If a user selects Tsukasa Domyouji and displays the single-focus character network with the entire link depth mode, all characters directly related from Tsukasa Domyouji are linked first, and then Sakurako Sanjo, Kiyonosuke Amakusa, and Shizuka Todou are linked from Tsukushi Makino, who is the most related to Tsukasa Domyouji.

If users wanted to find a certain story or character, networks for frequent co-occurrence characters using relatedness are useful. Figure 10 shows an example of a user inputting 100 as threshold value for relatedness and displaying frequent co-occurrence character networks. The screen shows characters whose relatedness is over 100. In this case, there is only one network centering on the main character Tsukushi Makino. The user can recall a side story between Tsukushi Makino and Tama, or another story between Yuuki Matsuoka and Sojirou Nishikado. The users can input threshold values as they like, enabling them to check the story in depth.

A variety of other networks are possible, and these should be investigated in the future.

6 Conclusions

To find and understand comic contents, we presented an approach of creating character networks with semantic relations from comics using the panels and speech balloons. First, we build a dataset that contains a comic’s characters and its words in speech balloons separated by the panels. Second, we extract the semantic relations among characters using the words in the speech balloons and panels. Third, we develop character networks by identifying the important characters using the frequencies of characters and their relatedness, and then we added semantic relations to these networks. As an initial step in finding semantic relations, we focused on kinship relations.

We implemented our approach and successfully built overview character networks that display important characters to overview the comic’s contents in a single diagram. This was carried out for four popular Japanese comics: *Naruto*, *Dragon Ball*, *Boys Over Flowers*, and *Space Brothers*. Finally, we evaluated our kinship relation extraction algorithms and the created overview character networks.

Future work includes improving the extracting kinship relation algorithms, evaluating the created overview character networks in more detail, considering additional semantic relations such as friends and rivals, and creating different types of networks.

References

- [1] W. S. Wang, “Globalizing manga: From Japan to Hong Kong and beyond,” *Mechademia*, pp. 23–45, 2006.

- [2] J. J. Miller, “The last million-selling comic book in North America? It’s Batman vs. Pokémon for the title,” 2014. [Online]. Available: <http://blog.comichron.com/2014/05/batman-vs-pokemon-last-million-selling.html>
- [3] *Weekly Shōnen Jump*. Shueisha (in Japanese), 1993, vol. 43.
- [4] *Weekly Shōnen Jump*. Shueisha (in Japanese), 2014, vol. 50.
- [5] H. Murakami, R. Kyogoku, and H. Ueda, “Creating character connections from manga,” in *Proc. 3rd Int. Conf. Agents and Artificial Intelligence (ICAART 2011)*, vol. 1, 2011, pp. 677–680.
- [6] A. Toriyama, *Dragon Ball*. Shueisha (in Japanese), 1992, vol. 32.
- [7] Y. Nagaoka, R. Kyogoku, and H. Murakami, “Creating character connections from comics using frames and words in balloons,” in *Human Communication Group Symposium 2015*, 2015, pp. 12–13, (in Japanese).
- [8] H. Murakami, Y. Nagaoka, and R. Kyogoku, “Creating character networks from comics using frames and words in balloons,” in *Proc. 2018 7th IIAI Int. Congr. Advanced Applied Informatics (IIAI-AAI 2018)*, 2018, pp. 1–6.
- [9] T. Ueno and K. Kazama, “Method of creating character networks based on speech timing in comics,” in *Proc. 11th ARG SIG-WI2*, 2017, pp. 1–6, (in Japanese).
- [10] D. K. Elson, N. Dames, and K. R. McKeown, “Extracting social networks from literary fiction,” in *Proc. 48th Annu. Meeting Association for Computational Linguistics (ACL 2010)*, 2010, pp. 138–147.
- [11] H. Nishihara, “Extraction of relations among characters in literary texts,” Master’s Thesis, JAIST, 2015, (in Japanese).
- [12] D. Kokkinakis and M. Malm, “Character profiling in 19th century fiction,” in *Proc. Language Technologies for Digital Humanities and Cultural Heritage Workshop*, 2011, pp. 70–77.
- [13] D. Kokkinakis, M. Malm, J. Bergenmar, and A. Ighe, “Semantics in storytelling in Swedish fiction,” in *DATeCH ’14 Proc. 1st Int. Conf. on Digital Access to Textual Cultural Heritage 2014*, 2014, pp. 137–142.
- [14] J. Lee and C. Y. Yeung, “Extracting networks of people and places from literary texts,” in *Proc. 26th Pacific Asia Conf. Language, Information, and Computation (PACLIC 2012)*, 2012, pp. 209–218.
- [15] J. Goto, Y. Yagi, A. Aizawa, and S. Sekine, “Generation of correlation charts from TV programs based on anaphora resolution,” in *22nd Annu. Conf. Japanese Society for Artificial Intelligence*, 2008, (in Japanese).
- [16] A. Ramakrishna, V. Martinez, N. Malandrakis, K. Singla, and S. Narayanan, “Linguistic analysis of differences in portrayal of movie characters,” in *Proc. 55th Annu. Meeting Association for Computational Linguistics (ACL 2017)*, 2017, pp. 1669–1678.

- [17] Y. Matsuo, J. Mori, M. Hamasaki, K. Ishida, T. Nishimura, H. Takeda, K. Hashida, and M. Ishiduka, “Polyphonet: An advanced social network extraction system,” in *Proc. WWW2006*, 2006, pp. 397–406.
- [18] S.-B. Park, Y.-W. Kim, M. Uddin, and G.-S. Jo, “Character-Net: Character network analysis from video,” in *WI-IAT '09 Proc. 2009 IEEE/WIC/ACM Int. Joint Conf. Web Intelligence and Intelligent Agent Technology*, vol. 1, 2005, pp. 305–308.
- [19] K. Yuan, R. Ji, H. Yao, X. Sun, P. Xu, and X. Liu, “VisualCor System: Search actor correlations in TV series,” in *ICIMCS '09: Proc. 1st Int. Conf. Internet Multimedia Computing and Service*, 2009, pp. 213–218.
- [20] M. Matsushita, “The possibility of comic computing,” in *Proc. 2nd ARG WI2*, no. 15, 2013, (in Japanese).
- [21] M. Matsushita, “The past and future of comic computing,” in *Proc. 11th JSAI Interactive Information Access and Visual Mining*, 2015, pp. 12–19, (in Japanese).
- [22] X. Pang, Y. Cao, R. W. H. Lau, and A. B. Chan, “A robust panel extraction method for manga,” in *MM '14 Proc. 22nd ACM int. conf. Multimedia*, 2014, pp. 1125–1128.
- [23] Q. Guo, K. Kato, N. Sato, and Y. Hoshino, “An algorithm for extracting text strings from comic strips,” in *SIGGRAPH '06 ACM SIGGRAPH 2006 Research posters*, no. 76, 2006.
- [24] W.-T. Chu and W.-W. li, “Manga FaceNet: Face detection in manga based on deep neural network,” in *ICMR '17 Proc. 2017 ACM Int. Conf. Multimedia Retrieval*, no. 76, 2017, pp. 412–415.
- [25] A. Fujimoto, T. Ogawa, K. Yamamoto, Y. Matsui, T. Yamasaki, and K. Aizawa, “Manga109 dataset and creation of metadata,” in *MANPU '16 Proc. 1st Int. Workshop on coMics Analysis, Processing and Understanding*, no. 2, 2016.
- [26] M. Kishimoto, *Naruto*. Shueisha (in Japanese), 2009, vol. 46.
- [27] M. Kishimoto, *Naruto*. Shueisha (in Japanese), 2010, vol. 50.
- [28] A. Toriyama, *Dragon Ball (Kanzenban)*. Shueisha (in Japanese), 2004, vol. 29.
- [29] A. Toriyama, *Dragon Ball (Kanzenban)*. Shueisha (in Japanese), 2004, vol. 34.
- [30] Y. Kamio, *Boys Over Flowers*. Shueisha (in Japanese), 1997, vol. 16.
- [31] Y. Kamio, *Boys Over Flowers*. Shueisha (in Japanese), 1998, vol. 20.
- [32] C. Koyama, *Space Brothers*. Kodansha (in Japanese), 2008, vol. 1.
- [33] C. Koyama, *Space Brothers*. Kodansha (in Japanese), 2009, vol. 5.