

Coreference Resolution on Math Problem Text in Japanese

Takumi Ito and Takuya Matsuzaki and Satoshi Sato

Graduate School of Engineering, Nagoya University, Japan
{takumi_i,matuzaki,ssato}@nuee.nagoya-u.ac.jp

Abstract

This paper describes a coreference resolution system for math problem text. Case frame dictionaries and a math taxonomy are utilized for supplying domain knowledge. The system deals with various anaphoric phenomena beyond well-studied entity coreferences.

1 Introduction

There is a growing interest in the natural language processing of mathematical problem text (e.g., Mitra and Baral, 2016; Upadhyay et al., 2016; Matsuzaki et al., 2017) because it serves as a prototypical example of a natural language interface for intelligent systems. In this paper, we describe a coreference resolution system for math problem text. In order to solve a math problem, all the anaphoric expressions including those referring to a proposition, as well as those referring to an entity, must be correctly analyzed. Although much research has been done on coreference resolution for texts such as newspaper, few researches have been done on math problems text.

Previous studies mainly focused on the relation between pronouns or definite noun phrases and their antecedents, and the resolution of zero (i.e., omitted) arguments of verbs. For instance, Iida et al. (2016) investigated zero anaphoric resolution on news text in Japanese and they considered only the zero arguments of a predicate.

On the other hand, in math problems, the arguments of unsaturated nouns are often omitted. This phenomenon is called bridging anaphora (Clark, 1975). For instance, in the following passage:

There is a right triangle ABC with $\angle C = 90^\circ$.
Let the length of the hypotenuse (of ϕ) be c .

the argument of “hypotenuse” is omitted and

has to be identified as “triangle ABC ”. There are studies that focus on such phenomena in Japanese (Sasano and Kurohashi, 2009) and English (Gerber and Chai, 2010). Our task includes theirs, but we need to consider issues peculiar to math problems. For example, we need to be stricter in detecting a zero pronoun, as in:

$x^2 = 1$ has a positive solution *(of ϕ).

One may be tempted to posit the existence of a zero argument of “a positive solution” that refers to “ $x^2 = 1$ ”. The principle of compositionality suggests that “a positive solution of $x^2 = 1$ ” has “1” as its denotation, but “ $x^2 = 1$ has 1” does not make sense. That is, we must *not* supply the “omitted” argument to “a solution” in this sentence to derive a correct semantic representation, e.g., $\exists x(x^2 = 1 \wedge x > 0)$. No previous work, including Iida et al’s, has dealt with such “syntactically determined” zero arguments as far as we are aware of. It is partly because most previous work has focused on the resolution of zero arguments of a predicate and there is no Japanese predicate that syntactically determines the antecedent of its zero argument.

Table 1 shows the types and the numbers of anaphoric expressions in math problems. For the survey, we randomly chose 100 problems from the entrance exam problems of seven top-ranked national universities in Japan from 1990 to 2014 (EEP) and 88 problems from the national standardized test for university admission in the even years from 1998 to 2014 (NST).

Table 1 reveals that there are few cases where (definite) common nouns are used anaphorically. One reason is that a symbol is often assigned to an entity as soon as it is introduced in a problem, such as in: “Let C be the circle that is ...” and the entity is thereafter referred to by the symbol but not by

Table 1: Type of Anaphoric Expression

Type		Example	Count	
			NST	EEP
Demonstrative	Pronoun	<i>sore</i> “it”, <i>sorera</i> “they”	3	2
	Determiner	<i>kono</i> “this”, <i>korerano</i> “these”	28	14
Zero pronoun as argument of unsaturated noun		<i>(ϕ-no) hankei</i> “radius (of ϕ)”	121	61
Common Noun		<i>houteishiki</i> “(the) equation”	1	5
Conditional Demonstrative		<i>konotoki</i> “for this case”	57	33
Others		<i>ippou</i> “one”, <i>tahou</i> “the other”	29	16

a definite noun. On the other hand, the number of occurrences of zero pronouns is very large. However, in 69% of them, the antecedents are syntactically determined. We need to discern them from the others so that heuristics-based antecedent determination is *not* applied to them. We can also see there are many conditional demonstratives, which refer not to entities but to propositions. There are several previous works that address them, such as Bejan and Harabagiu (2014), but it is much less studied compared to entity coreference resolution.

In the rest of the paper, we describe our coreference resolution system (§2) and provide some experimental results (§3). We then explain typical remaining errors (§4).

2 Methods

Our system handles four types of anaphoric expressions: pronouns, demonstrative determiner + nouns, zero pronouns, and conditional demonstratives. The first two types are analyzed according to a basic processing flow (§2.1). Zero pronouns are handled by slightly modifying the basic flow (§2.2). Conditional demonstratives are processed differently than the others since they refer to a proposition but not to an entity (§2.3).

2.1 Basic Processing Flow

An overview of the coreference resolution system is shown in Fig. 1. The input is a math problem in which math expressions are encoded in MathML¹. The system first tags each math expression with a label indicating its semantic type based on the syntactic pattern of the MathML expression. The semantic types are categories of mathematical objects such as “integer”, “real number”, “circle”, “ellipse”, etc. We currently have 543 semantic

¹<https://www.w3.org/TR/MathML3/>

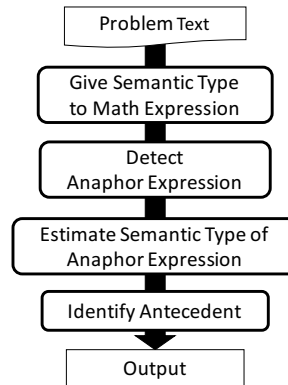


Figure 1: Overview of System

types. The distribution is very long-tailed and we observed only 106 of these types in the experiment presented in the current paper. For example, if the math expression is $\triangle ABC$, a label “triangle” is given. If the math expression is $y = 2x$, three labels, “function”, “equation”, and “line”, are given.

Next, anaphoric expressions are detected by a regular expression and the semantic types and the number of their antecedents are determined. If a noun appears immediately after a demonstrative such as in “this function”, the semantic type is determined by the noun. Otherwise, it is determined using a case frame dictionary. For example, in the case of “it intersects with the circle”, we know the semantic type of “it” is a type of “Shape” by the case frame of the predicate “intersect”.

The antecedents are then identified among the nouns, math expressions, and symbols that match the semantic type and are closest to the anaphoric expression. When a plural anaphoric expression includes a specific number (e.g., “these two triangles”), we identify as many antecedents as specified in all the preceding context from the closest to the anaphor. In contrast, for a plural anaphora including no specific number (e.g., “these triangles”), if we consider all the candidates in all the preceding context, there would be many false-positives. We thus need to limit the range of the context in which we identify the antecedents. Based on observation, we chose the window of two sentences as the range.

The case frame dictionary includes two sub-dictionaries, one for verbs and the other for unsaturated nouns. The verb case frame dictionary consists of 1858 frames and the unsaturated noun dictionary consists of 195 frames. They specify the semantic type and the number of arguments (sg: singular or pl: plural). An excerpt of each is shown in Fig. 2. The first example means “a

(Shape,sg-ga) (Point,sg-o) tooru
 [Shape-NOM Point-ACC pass]
 (Vec,sg-to) (Vec,sg-ga) chokkou-suru
 [Vec-COM Vec-NOM perpendicular]
 (Circle,sg-no) hankei
 [Circle,sg-GEN radius]
 (Integer,pl-no) kobaisu
 [Integer,pl-GEN common multiple]

Figure 2: Case Frame Dictionary (top: verbs, bottom: unsaturated nouns)

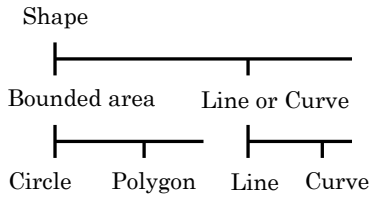


Figure 3: Math Taxonomy

planar figure passes through a point”, for instance.

We also use a math taxonomy to test if the type of an antecedent candidate is compatible with an anaphoric expression. A small part of the math taxonomy is shown in Fig. 3. The number of nodes of the taxonomy is 130.

2.2 Zero Pronoun

Processing of zero pronouns is almost the same as the basic processing flow. However, the detection of anaphoric expressions and the determination of their semantic types and plurality are done differently. In the detection of anaphoric expressions, firstly all the words in the dictionary of unsaturated nouns are extracted from a problem. The system judges that there is a zero argument of an unsaturated noun if none of the following conditions are met:

1. There is an overt genitive argument of the unsaturated noun that matches the semantic type of the argument (e.g., radius of a circle)
2. The unsaturated noun is in a relative clause and the noun modified by the relative clause matches the semantic type of the argument of the unsaturated noun
3. The unsaturated noun is an argument of a prescribed set of predicates, such as *motsu* “have” and *toru* “take”.

Case 2 and 3 exclude the cases where the antecedent of the zero argument is syntactically determined. A typical example of case 2 is *hankei-ga 3-no en* “a circle whose radius is 3”. In En-

glish, the relativizer “whose” specifies the genitive relation between “circle” and “radius” but in Japanese no such grammatical relations are specified by the (zero) relativizer. Therefore, the detection rule becomes complicated. Case 3 excludes the cases such as “ $f(x) = 0$ has a real solution (of ϕ)”, where the coreferential relation is controlled by the head verb (e.g., “has”) and hence the resolution is done in syntactic/semantic parsing stage.

Next, the semantic type and the number of the antecedent(s) are determined using the case frame dictionary. For example, when the unsaturated noun is “initial term (of ϕ)”, the type of the zero argument is assumed to be “number sequence” and the number is singular.

2.3 Conditional Demonstrative

In math problems, coreferential expressions such as “for this case”, “in case of (1)”, and “the following condition” refer to propositions. We call them *conditional demonstratives* in this paper. Our system handles three types of such expressions as described below.

First type is *kono/sono-toki* “for this/that case”. There are two usages of it. One indicates that all the previous conditions given in a problem are effective, and the other refers to a specific proposition, as in:

1. There is $\triangle ABC$ with $AB = BC = CA = 1$.	<u>Kono-toki,</u>	$\triangle ABC$ -no	menseki-o	motomeyo.
	<u>For this case</u>	$\triangle ABC$ -GEN	area-ACC	calculate.
<i>For this case, calculate the area of $\triangle ABC$.</i>				
2. $f(x)$ -no	saidaichi-to	<u>sono-toki-no</u>	x -o	
$f(x)$ -GEN	maximum-and	<u>for that case</u>	x -ACC	
motomeyo.				
find.				
<i>Find the maximum value of $f(x)$ and the value of x that gives the maximum</i>				

The underlined part, *Kono-toki* “for this case”, in the problem 1 indicates all the conditions given so far for $\triangle ABC$ have to be taken into consideration in solving it. As the English translation suggests, *Kono-toki* (and *Sono-toki*) in this usage can be omitted without changing the meaning of the problem. On the other hand, the underlined part *Sono-toki* “for that case” in the problem 2 points to a specific condition “ $f(x)$ becomes maximum” and cannot be omitted. We discriminate these two cases using regular expression patterns and rewrite the conditional demonstrative in the latter usage with the proposition it refers to (underlined in the example below):

$f(x)$ -no	saidaichi-to	$f(x)$ -ga	saidai-ni
$f(x)$ -GEN	maximum-COM	$f(x)$ -NOM	maximum
naru	toki-no	x -o	motomeyo.
become	when	x -ACC	find.

Find the maximum value of $f(x)$ and the value of x that gives the maximum

To identify what is referred to by a phrase such as “in the case of (1)”, we have to extract a proposition or a condition from the sub-problem designated by the problem number. To that end, we seek for a key phrase in the sub-problem that typically marks such a proposition. For example, *P-youna* “such that P”, *P-tame-no* “so that P”, *P-toki* “when P” and *P-to-suru* “assume that P” (P : proposition) are such key phrases. We rewrite the referring expression with the proposition identified by the key phrase as in:

- (1) $f(x)$ -no saidaichi-ga 3-ni naru you-ni a -o sadameyo.
 (1) Determine the value of a so that the maximum value of $f(x)$ is 3.
 (2) (1)-no baai-ni $f(x)$ -no saisyouchi-o motomeyo.
 (2) In the case of (1), find the minimum value of $f(x)$.
After rewriting (2)
 (2) $f(x)$ -no saidaichi-ga 3-ni naru-toki $f(x)$ -no saisyouchi-o motomeyo.
 (2) When the maximum value of $f(x)$ is 3, find the minimum value of $f(x)$.

For a referring expression such as *tsuginoshiki/joken* “the following formula/condition”, the referents are searched from the succeeding context. Specifically, the sentence or math expression just after the sentence including “the following condition” is identified as the referent and the referring expression is rewritten as follows:

- Tsugi-no houteishiki-o mitasu $\triangle ABC$ -o kangaeru :
 $\sin A = \sin B$
 Consider $\triangle ABC$ which satisfies the following equation : $\sin A = \sin B$
After rewriting
 $\sin A = \sin B$ -o mitasu $\triangle ABC$ -o kangaeru.
 Consider $\triangle ABC$ which satisfies $\sin A = \sin B$

3 Evaluation

We evaluated the system performance using three data sets. The first is 20 mock tests of Japanese national standardized test for university admission (MockNST) that consists of 74 problems. MockNST was also used for the system test while the development; hence it is a closed test data. The

Table 2: Accuracy of Zero Pronoun Detection

	Precision	Recall	F1
MockNST	74% (32/43)	84% (32/38)	79%
MockUT	89% (8/9)	100% (8/8)	93%

Table 3: Accuracy of antecedent identification

	Demonstrative	Zero Pronoun	Conditional	Total
MockNST	12/20	31/32	3/5	46/57 (81%)
MockUT	2/3	6/8	2/4	10/15 (67%)

second is 20 mock tests of the entrance exam of the University of Tokyo (MockUT) that consists of 41 problems. MockUT was kept unseen while the system development. We additionally used EEP data (see §1) for the evaluation of the resolution of conditional demonstratives.

Table 2 shows the accuracy of zero pronoun detection. The low precision means that unnecessary zero pronoun detection was performed. There were in total twelve such cases on MockNST and MockUT. Two thirds of them were due to an error in syntactic dependency analysis, which resulted in a failure in the recognition of the construction that determines the zero arguments of the unsaturated noun syntactically (§2.2). Table 3 shows the accuracy of the identification of antecedents.

Table 4 presents the accuracy at the problem level (i.e., perfect match) on MockUT. Approximately one third of the problems in MockUT include at least one coreference. About half of them could be completely resolved.

We also evaluated the accuracy of the resolution of conditional demonstratives on the 100 problems of EEP data set because conditional demonstratives were not frequently used in MockUT. In EEP, ten problems required coreference resolution of conditional demonstratives. Table 5 provides the details.

4 Remaining Problems

We found three types of frequent errors in the output of the current system. The first is due to an error in the dependency analysis, such as in:

- $x^4 - ax - a = 0$ -ga, kyojiku-jou-no fukusosuu-o kai-ni motsu youna jissuu a -o subete motomeyo.
 Find all real numbers a such that $x^4 - ax - a = 0$ has a solution on the imaginary axis of the complex plane.

In this problem, the dependency parser selected *motomeyo* “find” as the head of “ $x^2 - ax - a = 0$ ” but it should be *motsu* “has”. Due to this error,

Table 4: Evaluation at Problem level

#Coreferences	%Problems	Correct
≥ 1	32% (13/41)	54% (7/13)
None	68% (28/41)	—

Table 5: Evaluation of Conditional Demonstrative

	“for this case”	“in case of (1)”	Total
Accuracy	8/8	1/2	9/10 (90%)

the system cannot detect the pattern *X-ga Y-o kai-ni motsu* “X has a solution in Y” and unnecessary zero anaphora resolution was done.

The second is due to a kind of polymorphism in the natural language:

$\triangle ABC$ -no menseki-o $S(a)$ -de arawasu-toki kono-kansuu-no gurafu-o kake.
Letting $S(a)$ be the area of $\triangle ABC$, draw the graph of this function.

In this problem, $S(a)$ is first defined as a real number (area) but later referred to as a function. Due to this type mismatch, antecedent detection for *kono-kansuu* “this function” was failed.

The third is due to a failure in the recognition of the plurality of the zero pronoun.

En O -to houbutsusen C -ga ten $P(\sqrt{3}, 0)$ -o kyoyuu-shi, sarani P -ni okeru (ϕ -no) sessen-ga icchi-siteiru.
Circle O and parabola C share the point $P(\sqrt{3}, 0)$ and the tangent lines (of ϕ) at P coincide.

In this problem, a zero argument of *sessen* “tangent line” was successfully detected but it was wrongly interpreted as singular while it actually refers to “ O and C ”. Thus only C was identified as the antecedent. This is because there is no morphological indication of the number of a noun (such as -s in English) in Japanese. To solve this problem, we could utilize the case frame of *icchi-suru* “coincide”, which specifies a plural noun phrase as its subject. By this, we know *sessen* “tangent line(s)” actually signifies more than one entities and so is its zero argument.

5 Conclusion

This paper described a coreference resolution system for math problem text. The system deals with various anaphoric phenomena such as conditional demonstratives referring to propositions and syntactically controlled zero arguments. Evaluation showed the accuracy of the coreference resolution at the problem level was 54%. Our future work includes accurate recognition of the plurality of the

zero pronouns and fixing the error of the dependency analysis.

Acknowledgments

This research was supported by Todai Robot Project at National Institute of Informatics. We are gratefully acknowledge Gakko Hojin Takamiya Gakuen Yoyogi Seminar, Benesse Co., and Suken Shuppan Co., Ltd. for the math problem text data.

References

- Cosmin Adrian Bejan and Sanda Harabagiu. 2014. Un-supervised event coreference resolution. *Computational Linguistics* 40(2):311–347.
- Herbert H Clark. 1975. Bridging. In *Proceedings of the 1975 workshop on Theoretical issues in natural language processing*. Association for Computational Linguistics, pages 169–174.
- Matthew Gerber and Joyce Y Chai. 2010. Beyond nombank: A study of implicit arguments for nominal predicates. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 1583–1592.
- Ryu Iida, Kentaro Torisawa, Jong-Hoon Oh, Canasai Kruengkrai, and Julien Kloetzer. 2016. Intra-sentential subject zero anaphora resolution using multi-column convolutional neural network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. pages 1244–1254.
- Takuya Matsuzaki, Takumi Ito, Hidenao Iwane, Hirokazu Anai, and Noriko H. Arai. 2017. Semantic parsing of pre-university math problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 2131–2141.
- Arindam Mitra and Chitta Baral. 2016. Learning to use formulas to solve simple arithmetic problems. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. pages 2144–2153.
- Ryohei Sasano and Sadao Kurohashi. 2009. A probabilistic model for associative anaphora resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*. Association for Computational Linguistics, pages 1455–1464.
- Shyam Upadhyay, Ming-Wei Chang, Kai-Wei Chang, and Wen-tau Yih. 2016. Learning from explicit and implicit supervision jointly for algebra word problems. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. pages 297–306.