Is a 204 cm Man Tall or Small ?
Acquisition of Numerical Common Sense from the Web

Katsuma Narisawa[1] **Yotaro Watanabe**[1] **Junta Mizuno**[2]
Naoaki Okazaki[1,3] **Kentaro Inui**[1]
[1]Graduate School of Information Sciences, Tohoku University
[2]National Institute of Information and Communications Technology (NICT)
[3]Japan Science and Technology Agency (JST)
{katsuma,yotaro-w,junta-m,okazaki,inui}@ecei.tohoku.ac.jp

Abstract

This paper presents novel methods for modeling *numerical common sense*: the ability to infer whether a given number (e.g., *three billion*) is large, small, or normal for a given context (e.g., *number of people facing a water shortage*). We first discuss the necessity of numerical common sense in solving textual entailment problems. We explore two approaches for acquiring numerical common sense. Both approaches start with extracting numerical expressions and their *context* from the Web. One approach estimates the distribution of numbers co-occurring within a context and examines whether a given value is large, small, or normal, based on the distribution. Another approach utilizes textual patterns with which speakers explicitly expresses their judgment about the value of a numerical expression. Experimental results demonstrate the effectiveness of both approaches.

1 Introduction

Textual entailment recognition (RTE) involves a wide range of semantic inferences to determine whether the meaning of a hypothesis sentence (h) can be inferred from another text (t) (Dagan et al., 2006). Although several evaluation campaigns (e.g., PASCAL/TAC RTE challenges) have made significant progress, the RTE community recognizes the necessity of a deeper understanding of the core phenomena involved in textual inference. Such recognition comes from the ideas that crucial progress may derive from decomposing the complex RTE task into basic phenomena and from solving each basic phenomenon separately (Bentivogli et al., 2010; Sammons et al., 2010; Cabrio and Magnini, 2011; Toledo et al., 2012).

Given this background, we focus on solving one of the basic phenomena in RTE: semantic inference related to numerical expressions. The specific problem we address is acquisition of *numerical common sense*. For example,

(1) t : *Before long, 3b people will face a water shortage in the world.*
h : *Before long, a serious water shortage will occur in the world.*

Although recognizing the entailment relation between t and h is frustratingly difficult, we assume this inference is decomposable into three phases:

3b people face a water shortage.
$\Leftrightarrow$ *3,000,000,000 people face a water shortage.*
$\models$ *many people face a water shortage.*
$\models$ *a serious water shortage.*

In the first phase, it is necessary to recognize *3b* as a numerical expression and to resolve the expression *3b* into the exact amount *3,000,000,000*. The second phase is much more difficult because we need subjective but common-sense knowledge that *3,000,000,000 people* is a large number.

In this paper, we address the first and second phases of inference as an initial step towards semantic processing with numerical expressions. The contributions of this paper are four-fold.

1. We examine instances in existing RTE corpora, categorize them into groups in terms of the necessary semantic inferences, and discuss the impact of this study for solving RTE problems with numerical expressions.
2. We describe a method of normalizing numerical expressions referring to the same amount in text into a unified semantic representation.
3. We present approaches for aggregating numerical common sense from examples of numerical expressions and for judging whether a given amount is large, small, or normal.

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 382–391,
Sofia, Bulgaria, August 4-9 2013.

4. We demonstrate the effectiveness of this approach, reporting experimental results and analyses in detail. Although it would be ideal to evaluate the impact of this study on the overall RTE task, we evaluate each phase separately. We do this because the existing RTE data sets tend to exhibit very diverse linguistic phenomena, and it is difficult to employ such data for evaluating the real impact of this study.

2 Related work

Surprisingly, NLP research has paid little attention to semantic processing of numerical expressions. This is evident when we compare with temporal expressions, for which corpora (e.g., ACE-2005[1], TimeBank[2]) were developed with annotation schemes (e.g., TIMEX[3], TimeML[4]).

Several studies deal with numerical expressions in the context of information extraction (Bakalov et al., 2011), information retrieval (Fontoura et al., 2006; Yoshida et al., 2010), and question answering (Moriceau, 2006). Numbers such as product prices and weights have been common targets of information extraction. Fontoura et al. (2006) and Yoshida et al. (2010) presented algorithms and data structures that allow number-range queries for searching documents. However, these studies do not interpret the quantity (e.g., *3,000,000,000*) of a numerical expression (e.g., *3b people*), but rather treat numerical expressions as strings.

Banerjee et al. (2009) focused on quantity consensus queries, in which there is uncertainty about the quantity (e.g., *weight airbus A380 pounds*). Given a query, their approach retrieves documents relevant to the query and identifies the quantities of numerical expressions in the retrieved documents. They also proposed methods for enumerating and ranking the candidates for the consensus quantity intervals. Even though our study shares a similar spirit (modeling of consensus for quantities) with Banerjee et al. (2009), their goal is different: to determine ground-truth values for queries.

In question answering, to help "sanity check" answers with numerical values that were way out of common-sense ranges, IBM's PIQUANT (Prager et al., 2003; Chu-Carroll et al., 2003) used information in Cyc (Lenat, 1995). For example, their question-answering system rejects *200 miles* as a candidate answer for the *height of Mt. Everest*, since Cyc knows mountains are between 1,000 and 30,000 ft. high. They also consider the problem of variations in the precision of numbers (e.g., *5 million, 5.1 million, 5,200,390*) and unit conversions (e.g., *square kilometers* and *acres*).

Some recent studies delve deeper into the semantic interpretation of numerical expressions. Aramaki et al. (2007) focused on the physical size of an entity to predict the semantic relation between entities. For example, knowing that a *book* has a physical size of 20 cm $\times$ 25 cm and that a *library* has a size of 10 m $\times$ 10 m, we can estimate that a library contains a book (content-container relation). Their method acquires knowledge about entity size from the Web (by issuing queries like "*book (*cm x *cm)*"), and integrates the knowledge as features for the classification of relations.

Davidov and Rappoport (2010) presented a method for the extraction from the Web and approximation of numerical object attributes such as height and weight. Given an object-attribute pair, the study expands the object into a set of comparable objects and then approximates the numerical values even when no exact value can be found in a text. Aramaki et al. (2007) and Davidov and Rappoport (2010) rely on hand-crafted patterns (e.g., "*Object is * [unit] tall*"), focusing on a specific set of numerical attributes (e.g., height, weight, size). In contrast, this study can handle any kind of target and situation that is quantified by numbers, e.g., number of people facing a water shortage.

Recently, the RTE community has started to pay some attention to the appropriate processing of numerical expressions. Iftene (2010) presented an approach for matching numerical ranges expressed by a set of phrases (e.g., *more than* and *at least*). Tsuboi et al. (2011) designed hand-crafted rules for matching intervals expressed by temporal expressions. However, these studies do not necessarily focus on semantic processing of numerical expressions; thus, these studies do not normalize units of numerical expressions nor make inferences with numerical common sense.

Sammons et al. (2010) reported that most systems submitted to RTE-5 failed on examples

[1] http://www.itl.nist.gov/iad/mig/tests/ace/ace05/

[2] http://www.timeml.org/site/timebank/timebank.html

[3] http://timex2.mitre.org/

[4] http://timeml.org/site/index.html

where numeric reasoning was necessary. They argued the importance of aligning numerical quantities and performing numerical reasoning in RTE. LoBue and Yates (2011) identified 20 categories of common-sense knowledge that are prevalent in RTE. One of the categories comprises arithmetic knowledge (including computations, comparisons, and rounding). They concluded that many kinds of the common-sense knowledge have received scarce attention from researchers even though the knowledge is essential to RTE. These studies provided a closer look at the phenomena involved in RTE, but they did not propose a solution for handling numerical expressions.

3 Investigation of textual-entailment pairs with numerical expressions

In this section, we investigate textual entailment (TE) pairs in existing corpora in order to study the core phenomena that establish an entailment relation. We used two Japanese TE corpora: RITE (Shima et al., 2011) and Odani et al. (2008). RITE is an evaluation workshop of textual entailment organized by NTCIR-9, and it targets the English, Japanese, and Chinese languages. We used the Japanese portions of the development and training data. Odani et al. (2008) is another Japanese corpus that was manually created. The total numbers of text-hypothesis (T-H) pairs are 1,880 (RITE) and 2,471 (Odani).

We manually selected sentence pairs in which one or both of the sentences contained a numerical expression. Here, we define the term *numerical expression* as an expression containing a number or quantity represented by a numeral and a unit. For example, *3 kilometers* is a numerical expression with the numeral *3* and the unit *kilometer*. Note that *intensity of 4* is not a numerical expression because *intensity* is not a unit.

We obtained 371 pairs from the 4,351 T-H pairs. We determined the inferences needed to prove ENTAILMENT or CONTRADICTION of the hypotheses, and classified the 371 pairs into 11 categories. Note that we ignored T-H pairs in which numerical expressions were unnecessary to prove the entailment relation (e.g., *Socrates was sentenced to death by 500 jury members* and *Socrates was sentenced to death*). Out of 371 pairs, we identified 114 pairs in which numerical expressions played a central role in the entailment relation.

Table 1 summarizes the categories of TE phenomena we found in the data set. The largest category is *numerical matching* (32 pairs). We can infer an entailment relation in this category by aligning two numerical expressions, e.g., *2.2 million* $\models$ *over 800 thousand*. This is the most fundamental task in numerical reasoning, interpreting the amount (number, unit, and range) in a numerical expression. We address this task in Section 4.1. The second largest category requires common sense about numerical amounts. In order to recognize textual entailment of pairs in this category, we need common-sense knowledge about humans' subjective judgment of numbers. We consider this problem in Section 5.

To summarize, this study covers 37.9% of the instances in Table 1, focusing on the first and second categories. Due to space limitations, we omit the explanations for the other phenomena, which require such things as lexical knowledge, arithmetic operations, and counting. The coverage of this study might seem small, but it is difficult to handle varied phenomena with a unified approach. We believe that this study forms the basis for investigating other phenomena of numerical expressions in the future.

4 Collecting numerical expressions from the Web

In this paper, we explore two approaches to acquiring numerical common sense. Both approaches start with extracting numerical expressions and their *context* from the Web. We define a *context* as the verb and its arguments that appear around a numerical expression.

For instance, the context of *3b people* in the sentence *3b people face a water shortage* is "face" and "water shortage." In order to extract and aggregate numerical expressions in various documents, we converted the numerical expressions into semantic representations (to be described in Section 4.1), and extracted their context (to be described in Section 4.2).

The first approach for acquiring numerical common sense estimates the distribution of numbers that co-occur within a context, and examines whether a given value is large, small, or normal based on that distribution (to be described in Section 5.1). The second approach utilizes textual patterns with which speakers explicitly expresses their judgment about the value of a numerical ex-

Category	Definition	Example	#
Numerical matching	Aligning numerical expressions in T and H, considering differences in unit, range, etc.	t: It is said that there are about 2.2 million alcoholics in the whole country. h: It is estimated that there are over 800 thousand people who are alcoholics.	32
Numerical common sense	Inferring by interpreting the numerical amount (large or small).	t: In the middle of the 21st century, 7 billion people, corresponding to 70% of the global population, will face a water shortage. h: It is concerning that a serious water shortage will spread around the world in the near future.	12
Lexical knowledge	Inferring by using numerical aspects of word meanings.	t: Mr. and Ms. Sato celebrated their 25th wedding anniversary. h: Mr. and Ms. Sato celebrated their silver wedding anniversary.	12
Arithmetic	Arithmetic operations including addition and subtraction.	t: The number of 2,000-yen bills in circulation has increased to 450 million, in contrast with 440 million 5,000-yen bills. h: The number of 2,000-yen bills in circulation exceeds the number of 5,000-yen bills by 10 million bills.	11
Numeric-range expression of verbs	Numerical ranges expressed by verbs (e.g., *exceed*).	t: It is recorded that the maximum wave height reached 13.8 meters during the Sea of Japan Earthquake Tsunami in May 1983. h: During the Sea of Japan Earthquake, the height of the tsunami exceeded 10 meters.	9
Simple Rewrite Rule	This includes various simple rules for rewriting.	t: The strength of Taro's grip is No. 1 in his class. h: Taro's grip is the strongest in his class.	7
State change	Expressing the change of a value by a multiplier or ratio.	t: Consumption of pickled plums is 1.5 times the rate of 20 years ago. h: Consumption of pickled plums has increased.	6
Ordinal numbers	Inference by interpreting ordinal numbers.	t: Many precious lives were sacrificed in the Third World War. h: So far, there have been at least three World Wars.	6
Temporal expression	Inference by interpreting temporal expressions such as anniversary, age, and ordinal numbers.	t: Mr. and Ms. Sato celebrate their 25th wedding anniversary. h: Mr. and Ms. Sato got married 25 years ago.	3
Count	Counting up the number of various entities.	t: In Japan, there are the Asian Triopsidae, the American Triopsidae, and the European Triopsidae. h: In Japan, there are 3 types of Triopsidae.	3
Others			15
All			116

Table 1: Frequency and simple definitions for each category of the entailment phenomena in the survey.

Numerical Expression	Semantic representation		
	Value	Unit	Mod.
about seven grams	7	g	about
roughly 7 kg	7000	g	about
as heavy as 7 tons	7×10^6	g	large
as cheap as $1	1	$	small
30–40 people	$[30, 40]$	nin (*people*)	
more than 30 cars	30	dai (*cars*)	over
7 km per hour	7000	m/h	

Table 2: Normalized representation examples

String	Operation
gram(s)	set-unit: 'g'
kilogram(s)	set-unit: 'g'; multiply-value: 1,000
kg	set-unit: 'g'; multiply-value: 1,000
ton(s)	set-unit: 'g'; multiply-value: 1,000,000
nin (*people*)	set-unit: 'nin' (*person*)
about	set-modifier: 'about'
as many as	set-modifier: 'large'
as little as	set-modifier: 'small'

Table 3: An example of unit/modifier dictionary

pression (to be explained in Section 5.2).

In this study, we acquired numerical common sense from a collection of 8 billion sentences in 100 million Japanese Web pages (Shinzato et al., 2012). For this reason, we originally designed text patterns specialized for Japanese dependency trees. For the sake of the readers' understanding, this paper uses examples with English translations for explaining language-independent concepts, and both Japanese and English translations for explaining language-dependent concepts.

4.1 Extracting and normalizing numerical expressions

The first step for collecting numerical expressions is to recognize when a numerical expression is mentioned and then to normalize it into a semantic representation. This is the most fundamental step in numerical reasoning and has a number of applications. For example, this step handles cases of *numerical matching*, as in Table 1.

The semantic representation of a numerical expression consists of three fields: the value or range of the real number(s)[5], the unit (a string), and the optional modifiers. Table 2 shows some examples of numerical expressions and their semantic representations. During normalization, we identified spelling variants (e.g., *kilometer* and *km*) and transformed auxiliary units into their corresponding canonical units (e.g., *2 tons* and *2,000 kg* to *2,000,000 grams*). When a numerical expression is accompanied by a modifier such as *over*, *about*, or *more than*, we updated the value and modifier fields appropriately.

[5]Internally, all values are represented by ranges (e.g., 75 is represented by the range $[75, 75]$).

We developed an extractor and a normalizer for Japanese numerical expressions[6]. We will outline the algorithm used in the normalizer with an example sentence: "Roughly three thousand kilograms of meats have been provided every day."

1. Find numbers in the text by using regular expressions and convert the non-Arabic numbers into their corresponding Arabic numbers. For example, we find *three thousand*[7] and represent it as $3,000$.
2. Check whether the words that precede or follow the number are units that are registered in the dictionary. Transform any auxiliary units. In the example, we find that *kilograms*[8] is a unit. We multiply the value $3,000$ by $1,000$, and obtain the value $3,000,000$ with the unit g.
3. Check whether the words that precede or follow the number have a modifier that is registered in the dictionary. Update the value and modifier fields if necessary. In the example, we find *roughly* and set *about* in the modifier field.

We used a dictionary[9] to perform procedures 2 and 3 (Table 3). If the words that precede or follow an extracted number match an entry in the dictionary, we change the semantic representation as described in the operation.

The modifiers 'large' and 'small' require elaboration because the method in Section 5.2 relies heavily on these modifiers. We activated the modifier 'large' when a numerical expression occurred with the Japanese word *mo*, which roughly corresponds to *as many as*, *as large as*, or *as heavy as* in English[10]. Similarly, we activated the modifier 'small' when a numerical expression occurred with the word *shika*, which roughly corresponds to *as little as*, *as small as*, or *as light as*[11]. These modifiers are important for this study, reflecting the writer's judgment about the amount.

[6]The software is available at `http://www.cl.ecei.tohoku.ac.jp/~katsuma/software/normalizeNumexp/`

[7]In Japanese $3,000$ is denoted by the Chinese symbols "三千".

[8]We write kilograms as "キログラム" in Japanese.

[9]The dictionary is bundled with the tool. See Footnote 6.

[10]In Japanese, we can use the word *mo* with a numerical expression to state that the amount is 'large' regardless of how large it is (e.g., large, big, many, heavy).

[11]Similarly, we can use the word *shika* with any adjective.

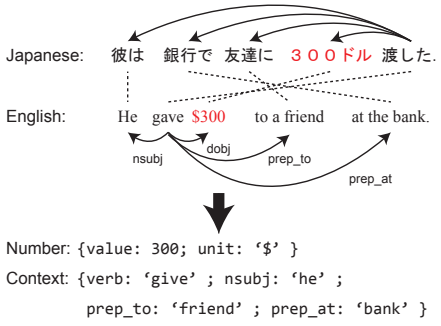

Figure 1: Example of context extraction

4.2 Extraction of context

The next step in acquiring numerical common sense is to capture the context of numerical expressions. Later, we will aggregate numbers that share the same context (see Section 5). The context of a numerical expression should provide sufficient information to determine what it measures. For example, given the sentence, "He gave $300 to a friend at the bank," it would be better if we could generalize the context to *someone gives money to a friend* for the numerical expression *$300*. However, it is a nontrivial task to design an appropriate representation of varying contexts. For this reason, we employ a simple rule to capture the context of numerical expressions: we represent the context with the verb that governs the numerical expression and its typed arguments.

Figure 1 illustrates the procedure for extracting the context of a numerical expression[12]. The component in Section 4.1 recognizes *$300* as a numerical expression, then normalizes it into a semantic representation. Because the numerical expression is a dependent of the verb *gave*, we extract the verb and its arguments (except for the numerical expression itself) as the context. After removing inflections and function words from the arguments, we obtain the context representation of Figure 1.

5 Acquiring numerical common sense

In this section, we present two approaches for acquiring numerical common sense from a collection of numerical expressions and their contexts. Both approaches start with collecting the numbers (in semantic representation) and contexts of numerical expressions from a large number of sentences (Shinzato et al., 2012), and storing them

[12]The English dependency tree might look peculiar because it is translated from the Japanese dependency tree.

in a database. When a context and a value are given for a prediction (hereinafter called the query context and query value, respectively), these approaches judge whether the query value is large, small, or normal.

5.1 Distribution-based approach

Given a query context and query value, this approach retrieves numbers associated with the query context and draws a distribution of normalized numbers. This approach considers the distribution estimated for the query context and determines if the value is within the top 5 percent (large), within the bottom 5 percent (small), or is located in between these regions (normal).

The underlying assumption of this approach is that the real distribution of a query (e.g., *money given to a friend*) can be approximated by the distribution of numbers co-occurring with the context (e.g., *give* and *friend*) on the Web. However, the context space generated in Section 4.2 may be too sparse to find numbers in the database, especially when a query context is fine-grained. Therefore, when no item is retrieved for the query context, we employ a backoff strategy to drop some of the uninformative elements in the query context: elements are dropped from the context based on the type of argument, in this order: *he* (prep_to), *kara* (prep_from), *ha* (nsubj), *yori* (prep_from), *made* (prep_to), *nite* (prep_at), *de* (prep_at, prep_by), *ni* (prep_at), *wo* (dobj), *ga* (nsubj), and verb.

5.2 Clue-based approach

This approach utilizes textual clues with which a speaker explicitly expresses his or her judgment about the amount of a numerical expression. We utilize large and small modifiers (described in Section 4.1), which correspond to textual clues *mo (as many as, as large as)* and *shika (only, as few as)*, respectively, for detecting humans' judgments. For example, we can guess that *$300* is *large* if we find an evidential sentence[13], *He gave as much as $100 to a friend.*

Similarly to the distribution-based approach, this approach retrieves numbers associated with the query context. This approach computes the largeness $L(x)$ of a value x:

$$L(x) = \frac{p_l(x)}{p_s(x) + p_l(x)}, \quad (1)$$

$$p_l(x) = \frac{\left|\{r | r_v < x \wedge r_m \ni \textit{large}\}\right|}{\left|\{r | r_m \ni \textit{large}\}\right|}, \quad (2)$$

$$p_s(x) = \frac{\left|\{r | r_v > x \wedge r_m \ni \textit{small}\}\right|}{\left|\{r | r_m \ni \textit{small}\}\right|}. \quad (3)$$

In these equations, r denotes a retrieved item for the query context, and r_v and r_m represent the normalized value and modifier flags, respectively, of the item r. The numerator of Equation 2 counts the number of numerical expressions that support the judgment that x is large[14], and its denominator counts the total number of numerical expressions with *large* as a modifier. Therefore, $p_l(x)$ computes the ratio of times there is textual evidence that says that x is large, to the total number of times there is evidences with *large* as a modifier. In an analogous way, $p_s(x)$ is defined to be the ratio for evidence that says x is small. Hence, $L(x)$ approaches 1 if everyone on the Web claims that x is large, and approaches 0 if everyone claims that x is small. This approach predicts *large* if $L(x) > 0.95$, *small* if $L(x) < 0.05$, and *normal* otherwise.

6 Experiments

6.1 Normalizing numerical expressions

We evaluated the method that we described in Section 4.1 for extracting and normalizing numerical expressions. In order to prepare a gold-standard data set, we obtained 1,041 sentences by randomly sampling about 1% of the sentences containing numbers (Arabic digits and/or Chinese numerical characters) in a Japanese Web corpus (100 million pages) (Shinzato et al., 2012). For every numerical expression in these sentences, we manually determined a tuple of the normalized value, unit, and modifier. Here, non-numerical expressions such as temporal expressions, telephone numbers, and postal addresses, which were very common, were beyond the scope of the project[15]. We obtained 329 numerical expressions from the 1,041 sentences.

We evaluated the correctness of the extraction and normalization by measuring the precision and

[13] Although the sentence states a judgment about $100, we can infer that $300 is also large because $300 > $100.

[14] This corresponds to the events where we find an evidence expression "as many as r_v", where $r_v < x$.

[15] If a tuple was extracted from a non-numerical expression, we regarded this as a false positive

recall using the gold-standard data set[16]. Our method performed with a precision of 0.78 and a recall of 0.92. Most of the false negatives were caused by the incompleteness of the unit dictionary. For example, the proposed method could not identify *1Ghz* as a numerical expression because the unit dictionary did not register *Ghz* but *GHz*. It is trivial to improve the recall of the method by enriching the unit dictionary.

The major cause of false positives was the semantic ambiguity of expressions. For example, the proposed method identified *Seven Hills* as a numerical expression although it denotes a location name. In order to reduce false positives, it may be necessary to utilize broader contexts when locating numerical expressions; this could be done by using, for example, a named entity recognizer. This is the next step to pursue in future work.

However, these errors do not have a large effect on the estimation of the distribution of the numerical values that occur with specific named entities and idiomatic phrases. Moreover, as explained in Section 5, we draw distributions for fine-grained contexts of numerical expressions. For these reasons, we think that the current performance is sufficient for acquiring numerical common sense.

6.2 Acquisition of numerical common sense

6.2.1 Preparing an evaluation set

We built a gold-standard data set for numerical common sense. We applied the method in Section 4.1 to sentences sampled at random from the Japanese Web corpus (Shinzato et al., 2012), and we extracted 2,000 numerical expressions. We asked three human judges to annotate every numerical expression with one of six labels, *small*, *relatively small*, *normal*, *relatively large*, *large*, and *unsure*. The label *relatively small* could be applied to a numerical expression when the judge felt that the amount was rather small (below the normal) but hesitated to label it *small*. The label *relatively large* was defined analogously. We gave the following criteria for labeling an item as *unsure*: when the judgment was highly dependent on the context; when the sentence was incomprehensible; and when it was a non-numerical expressions (false positives of the method are discussed in Section 4.1).

Table 4 reports the inter-annotator agreement.

[16] All fields (value, unit, modifier) of the extracted tuple must match the gold-standard data set.

Agreement	# expressions
3 annotators	735 (36.7%)
2 annotators	963 (48.2%)
no agreement	302 (15.1%)
Total	2000 (100.0%)

Table 4: Inter-annotator agreement

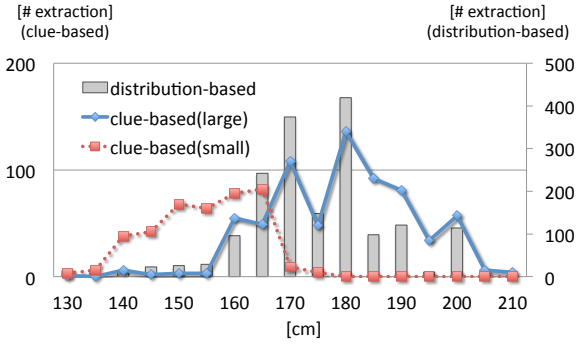

Figure 2: Distributions of numbers with large and small modifiers for the context *human's height*.

For the evaluation of numerical expressions in the data set, we used those for which at least two annotators assigned the same label. After removing the *unsure* instances, we obtained 640 numerical expressions (20 *small*, 35 *relatively small*, 152 *normal*, 263 *relatively large*, and 170 *large*) as the evaluation set.

6.2.2 Results

The proposed method extracted about 23 million pairs of numerical expressions and their context from the corpus (with 100 million Web pages). About 15% of the extracted pairs were accompanied by either a large or small modifier. Figure 2 depicts the distributions of the context *human's height* produced by the distribution-based and clue-based approaches. These distributions are quite reasonable as common-sense knowledge: we can interpret that numbers under 150 cm are perceived as small and those above 180 cm as large.

We measured the correctness of the proposed methods on the gold-standard data. For this evaluation, we employed two criteria for correctness: strict and lenient. With the strict criterion, the method must predict a label identical to that in the gold-standard. With the lenient criterion, the method was also allowed to predict either *large/small* or *normal* when the gold-standard label was *relatively large/small*.

Table 5 reports the precision (P), recall (R), F1 (F1), and accuracy (Acc) of the proposed methods.

No.	System	Gold	Sentence	Remark
1	small	small	I think that three men can create such a great thing in the world.	Correct
2	normal	normal	I have two cats.	Correct
3	large	large	It's above 32 centigrade.	Correct
4	large	large	I earned 10 million yen from horse racing.	Correct
5	small	normal	There are 2 reasons.	*Difficulty in judging small*. Since a few people say, "There are only 2 reasons," our approach predicted a *small* label.
6	small	large	Ten or more people came, and my eight-mat room was packed.	*Difficulty in modeling the context* because this sentence omits the locational argument for the verb *came*. We should extract the context as *the number of people who came to my eight-mat room* instead of *the number of people who came*.
7	small	normal	I have two friends who have broken up with their boyfriends recently.	*Difficulty in modeling the context.* We should extract context as *the number of friends who have broken up with their boyfriends recently* instead of *the number of friends*.
8	small	large		*Lack of knowledge.* We extract the context as *the number of heads of a turtle*, but no corresponding information was found on the Web.

Table 6: Output example and error analysis. We present translations of the sentences, which were originally in Japanese.

Approach	Label	P	R	F1	Acc
Distribution	large+	0.892	0.498	0.695	0.760
	normal+	0.753	0.935	0.844	
	small+	0.273	0.250	0.262	
Distribution	large	0.861	0.365	0.613	0.590
	normal	0.529	0.908	0.719	
	small	0.222	0.100	0.161	
Clue	large+	0.923	0.778	0.851	0.770
	normal+	0.814	0.765	0.790	
	small+	0.228	0.700	0.464	
Clue	large	0.896	0.659	0.778	0.620
	normal	0.593	0.586	0.590	
	small	0.164	0.550	0.357	

Table 5: Precision (P), recall (R), F1 score (F1), and accuracy (Acc) of the acquisition of numerical common sense.

Labels with the suffix '+' correspond to the lenient criterion. The clue-based approach achieved 0.851 F1 (for large), 0.790 F1 (for normal), and 0.464 (for small) with the lenient criterion. The performance is surprisingly good, considering the subjective nature of this task.

The clue-based approach was slightly better than the distribution-based approach. In particular, the clue-based approach is good at predicting large and small labels, whereas the distribution-based approach is good at predicting normal labels. We found some targets for which the distribution on the Web is skewed from the 'real' distribution. For example, let us consider the distribution of the context *"the amount of money that a person wins in a lottery"*. We can find a number of sentences like *if you won the 10-million-dollar lottery, ...* In other words, people talk about a large amount of money even if they did not win any money at all. In order to remedy this problem, we may need to enrich the context representation by introducing, for example, the factuality of an event.

6.2.3 Discussion

Table 6 shows some examples of predictions from the clue-based approach. Because of space limitations, we mention only the false instances of this approach.

The clue-based approach tends to predict *small* even if the gold-standard label is *normal*. About half of the errors of the clue-based approach were of this type; this is why the precision for *small* and the recall for *normal* are low. The cause of this error is exemplified by the sentence, "there are two reasons." Human judges label *normal* to the numerical expression *two reasons*, but the method predicts *small*. This is because a few people say *there are only two reasons*, but no one says *there are as many as two reasons.* In order to handle these cases, we may need to incorporate the distribution information with the clue-based approach.

We found a number of examples for which modeling the context is difficult. Our approach represents the context of a numerical expression with the verb that governs the numerical expression and its typed arguments. However, this approach sometimes misses important information, especially when an argument of the verb is omitted (Example 6). The approach also suffers from the relative clause in Example 7, which conveys an essential context of the number. These are similar to the scope-ambiguity problem such as encoun-

tered with negation and quantification; it is difficult to model the scope when a numerical expression refers to a situation.

Furthermore, we encountered some false examples even when we were able to precisely model the context. In Example 8, the proposed method was unable to predict the label correctly because no corresponding information was found on the Web. The proposed method might more easily predict a label if we could generalize the word *turtle* as *animal*. It may be worth considering using language resources (e.g., WordNet) to generalize the context.

7 Conclusions

We proposed novel approaches for acquiring numerical common sense from a collection of texts. The approaches collect numerical expressions and their contexts from the Web, and acquire numerical common sense by considering the distributions of normalized numbers and textual clues such as *mo (as many as) and shika (only, as few as)*. The experimental results showed that our approaches can successfully judge whether a given amount is large, small, or normal. The implementations and data sets used in this study are available on the Web[17]. We believe that acquisition of numerical common sense is an important step towards a deeper understanding of inferences with numbers.

There are three important future directions for this research. One is to explore a more sophisticated approach for precisely modeling the contexts of numbers. Because we confirmed in this paper that these two approaches have different characteristics, it would be interesting to incorporate textual clues into the distribution-based approach by using, for example, machine learning techniques. Finally, we are planning to address the 'third phase' of the example explained in Section 1: associating *many people face a water shortage* with *a serious water shortage*.

Acknowledgments

This research was partly supported by JST, PRESTO. This research was partly supported by JSPS KAKENHI Grant Numbers 23240018 and 23700159.

[17] http://www.cl.ecei.tohoku.ac.jp/~katsuma/resource/numerical_common_sense/

References

Eiji Aramaki, Takeshi Imai, Kengo Miyo, and Kazuhiko Ohe. 2007. Uth: Svm-based semantic relation classification using physical sizes. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 464–467.

Anton Bakalov, Ariel Fuxman, Partha Pratim Talukdar, and Soumen Chakrabarti. 2011. SCAD: collective discovery of attribute values. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 447–456.

Somnath Banerjee, Soumen Chakrabarti, and Ganesh Ramakrishnan. 2009. Learning to rank for quantity consensus queries. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 243–250.

Luisa Bentivogli, Elena Cabrio, Ido Dagan, Danilo Giampiccolo, Medea Lo Leggio, and Bernardo Magnini. 2010. Building textual entailment specialized data sets: a methodology for isolating linguistic phenomena relevant to inference. *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, pages 3542–3549.

Elena Cabrio and Bernardo Magnini. 2011. Towards component-based textual entailment. In *Proceedings of the Ninth International Conference on Computational Semantics*, IWCS '11, pages 320–324.

Jennifer Chu-Carroll, David A. Ferrucci, John M. Prager, and Christopher A. Welty. 2003. Hybridization in question answering systems. In *New Directions in Question Answering'03*, pages 116–121.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190.

Dmitry Davidov and Ari Rappoport. 2010. Extraction and approximation of numerical attributes from the web. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1308–1317.

Marcus Fontoura, Ronny Lempel, Runping Qi, and Jason Zien. 2006. Inverted index support for numeric search. *Internet Mathematics*, 3(2):153–185.

Adrian Iftene and Mihai-Alex Moruz. 2010. UAIC participation at RTE-6. In *Proceedings of the Third Text Analysis Conference (TAC 2010) November*.

Douglas B Lenat. 1995. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38.

Peter LoBue and Alexander. Yates. 2011. Types of common-sense knowledge needed for recognizing

textual entailment. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 329–334.

Véronique Moriceau. 2006. Generating intelligent numerical answers in a question-answering system. In *Proceedings of the Fourth International Natural Language Generation Conference*, INLG '06, pages 103–110.

Michitaka Odani, Tomohide Shibata, Sadao Kurohashi, and Takayuki Nakata. 2008. Building data of japanese text entailment and recognition of inferencing relation based on automatic achieved similar expression. *In Proceeding of 14th Annual Meeting of the Association for "atural Language Processing*, pages 1140–1143.

John M. Prager, Jennifer Chu-Carroll, Krzysztof Czuba, Christopher A. Welty, Abraham Ittycheriah, and Ruchi Mahindru. 2003. IBM's PIQUANT in TREC2003. In *TREC*, pages 283–292.

Mark Sammons, Vinod V.G. Vydiswaran, and Dan Roth. 2010. Ask not what textual entailment can do for you... In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1199–1208.

Hideki Shima, Hiroshi Kanayama, Cheng-Wei Lee, Chuan-Jie Lin, Teruko Mitamura, Yusuke Miyao, Shuming Shi, and Koichi Takeda. 2011. Overview of ntcir-9 rite: Recognizing inference in text. In *Proceeding of NTCIR-9 Workshop Meeting*, pages 291–301.

Keiji Shinzato, Tomohide Shibata, Daisuke Kawahara, and Sadao Kurohashi. 2012. Tsubaki: An open search engine infrastructure for developing information access methodology. *Journal of Information Processing*, 20(1):216–227.

Assaf Toledo, Sophia Katrenko, Stavroula Alexandropoulou, Heidi Klockmann, Asher Stern, Ido Dagan, and Yoad Winter. 2012. Semantic annotation for textual entailment recognition. In *Proceedings of the 11th Mexican International Conference on Artificial Intelligence*, MICAI '12.

Yuta Tsuboi, Hiroshi Kanayama, Masaki Ohno, and Yuya Unno. 2011. Syntactic difference based approach for ntcir-9 rite task. In *Proceedings of the 9th NTCIR Workshop*, pages 404–411.

Minoru Yoshida, Issei Sato, Hiroshi Nakagawa, and Akira Terada. 2010. Mining numbers in text using suffix arrays and clustering based on dirichlet process mixture models. *Advances in Knowledge Discovery and Data Mining*, pages 230–237.