Logical Inference for Counting on Semi-structured Tables

Tomoya Kurosawa and **Hitomi Yanaka**
The University of Tokyo
{kurosawa-tomoya, hyanaka}@is.s.u-tokyo.ac.jp

Abstract

Recently, the Natural Language Inference (NLI) task has been studied for semi-structured tables that do not have a strict format. Although neural approaches have achieved high performance in various types of NLI, including NLI between semi-structured tables and texts, they still have difficulty in performing a numerical type of inference, such as counting. To handle a numerical type of inference, we propose a logical inference system for reasoning between semi-structured tables and texts. We use logical representations as meaning representations for tables and texts and use model checking to handle a numerical type of inference between texts and tables. To evaluate the extent to which our system can perform inference with numerical comparatives, we make an evaluation protocol that focuses on numerical understanding between semi-structured tables and texts in English. We show that our system can more robustly perform inference between tables and texts that requires numerical understanding compared with current neural approaches.

1 Introduction

Natural Language Inference (NLI) (Dagan et al., 2006) is one of the most fundamental tasks to determine whether a premise entails a hypothesis. Recently, researchers have developed benchmarks not only for texts but for other kinds of resources as well, a table being one example. Previous studies have targeted database-style structured tables (Pasupat and Liang, 2015; Wiseman et al., 2017; Krishnamurthy et al., 2017) and semi-structured tables, such as the infoboxes in Wikipedia (Lebret et al., 2016; Gupta et al., 2020). Our focus here is on the NLI task on semi-structured tables, where we handle a semi-structured table as a premise and a sentence as a hypothesis.

In Figure 1, for example, we consider the semi-structured table as a given premise and take *Joe Biden was born in November* as Hypothesis 1. We can conclude that Hypothesis 1 is entailed by the table. A semi-structured table has only two columns and describes a single object, which is indicated in the title. We call elements of the first column, such as **Political Party**, *keys*, each of which has an associated *value* in the second column such as *Democratic* (*1969–present*). Pairs of keys and values are called *rows*. It is relatively difficult to understand the information contained in infobox tables because (i) values are not limited to words or phrases, and sometimes whole sentences, and (ii) a row can contain more than one type of information, such as the birthday and birthplace in the **Born** row.

Joe Biden	
Born	Joseph Robinette Biden Jr. November 20, 1942 (age 79) Scranton, Pennsylvania, U.S.
Political party	Democratic (1969–present)
Spouse(s)	Neilia Hunter (m. 1966; died 1972) Jill Jacobs (m. 1977)

Hypothesis 1: Joe Biden was born in November.
Hypothesis 2: Joe Biden has had more than two wives.

Figure 1: A semi-structured table describing Joe Biden[1] and two hypothesis sentences. This table entails Hypothesis 1 and contradicts Hypothesis 2.

In recent years, modern neural network (NN) approaches have achieved high performance in many Natural Language Understanding benchmarks, such as BERT (Devlin et al., 2019). NN-based approaches (Neeraja et al., 2021) have also achieved high accuracy on the NLI task between semi-structured tables and texts, but previous studies have questioned whether NN-based models truly understand the various linguistic phenomena

[1]The table was retrieved from https://en.wikipedia.org/wiki/Joe_Biden on February 25, 2022. Some rows have been removed to save space.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Student Research Workshop, pages 84 - 96
May 22-27, 2022 ©2022 Association for Computational Linguistics

(Jia and Liang, 2017; Naik et al., 2018; Rozen et al., 2019; Ravichander et al., 2019; Richardson et al., 2020). These studies have shown that NN-based approaches have failed to achieve a high performance in numerical reasoning.

In this paper, we focus on a numerical type of inference on semi-structured tables, which requires understanding the number of items in a table as well as numerical comparisons. Numerical comparatives are among the more challenging linguistic phenomena that involve generalized quantifiers. For example, the phrase *more than* in Hypothesis 2 in Figure 1 is a numerical comparative and compares two and the number of wives. For dealing with numerical comparatives, Haruta et al. (2020a,b) achieved high performance by developing a logical inference system based on formal semantics. However, Haruta et al. (2020a,b) concentrated on the inference between texts only, and inference systems that reliably perform inference between tables and texts involving numerical comparatives have not yet been developed.

Thus, we aim to develop a logical inference system between semi-structured tables and texts, especially for numerical reasoning. While previous work (Pasupat and Liang, 2015; Wiseman et al., 2017; Krishnamurthy et al., 2017) has provided semantic parsers of constructing query languages such as SQLs for question answering on database-style tables, we present logical representations for semi-structured tables to enable a numerical type of inference on semi-structured tables. Furthermore, the existing NLI dataset for semi-structured tables (Gupta et al., 2020) does not contain sufficient test cases for understanding numerical comparatives. Thus, there is a need for an evaluation protocol that investigates the numerical reasoning skills of NLI systems for semi-structured tables.

Given this background, our main contributions in this paper are the following:

1. We propose a logical inference system for handling numerical comparatives that is based on formal semantics for NLI between semi-structured tables and texts.

2. We provide an evaluation protocol and dataset that focus on numerical comparatives between semi-structured tables and texts.

3. We demonstrate the increased performance of our inference system compared with previous NN models on the NLI dataset, focusing on numerical comparatives between semi-structured tables and texts.

Our system and dataset will be publicly available at `https://github.com/ynklab/sst_count`.

2 Related Work and Background

This section explains the related work of logic-based NLI approaches and the background of model checking, which is used for inference between semi-structured tables and sentences in our proposed system.

2.1 Logic-based Approach

Based on the analysis of formal semantics, logic-based NLI approaches can handle a greater variety of linguistic phenomena than NN-based approaches can. Some logic-based NLI approaches using syntactic and semantic parsers based on formal semantics have been proposed (Bos, 2008; Abzianidze, 2015; Mineshima et al., 2015; Hu et al., 2020; Haruta et al., 2020a,b). These logic-based approaches can derive semantic representations of sentences involving linguistically challenging phenomena, such as generalized quantifiers and comparatives, based on Combinatory Categorial Grammar (CCG) (Steedman, 2000) syntactic analysis. CCG is often used in these approaches because it has a tiny number of combinatory rules, which is suitable for semantic composition from syntactic structures. In addition, robust CCG parsers are readily available (Clark and Curran, 2007; Yoshikawa et al., 2017).

Regarding logic-based approaches for inference other than inference between texts, Suzuki et al. (2019) proposed a logical inference system for inference between images and texts. Their system converts images to first-order logic (FOL) structures by using image datasets where structured representations of the images are annotated. They then get FOL formulas P for images from these structures along with the associated image captions. Hypothesis sentences are translated into FOL formulas H through the use of a semantic parser (Martínez-Gómez et al., 2016). For inference, they used automated theorem proving and sought to prove $P \vdash H$. Our proposed inference system between semi-structured tables and texts is inspired by Suzuki et al. (2019). While the previous system uses automated theorem proving for in-

$\boldsymbol{D} = \{B_1, G_1, G_2\}$
$\boldsymbol{V} = \{(\text{ALICE}, \{G_1\}), (\text{BOB}, \{B_1\}), (\text{CATHY}, \{G_2\}),$
$(\text{BOY}, \{B_1\}), (\text{GIRL}, \{G_1, G_2\}),$
$(\text{LIKE}, \{(B_1, G_1), (B_1, G_2), (G_1, B_1)\})\}$

Logical formula	Output
$\exists x.\exists y.(\text{BOY}(x) \wedge \text{LIKE}(x, y))$	True
$\exists x.\exists y.(\text{GIRL}(x) \wedge \text{GIRL}(y) \wedge \text{LIKE}(x, y))$	False
$\exists x.\exists y.(\text{CAT}(y) \wedge \text{LIKE}(x, y))$	Undefined

Figure 2: Outputs of model checking based on an example model and three formulas.

ference between images and texts, our system uses model checking to judge whether a given text is true under a given table, and it is expected to be a faster method.

2.2 Model Checking

We use model checking in the Natural Language Toolkit (NLTK) (Bird and Loper, 2004; Garrette and Klein, 2009) for making inference between tables and texts. This system judges a truth-value of an FOL formula based on FOL structures. An FOL structure (called *model*) is defined by a pair of the domain $\boldsymbol{D}$ and the valuation $\boldsymbol{V}$, where $\boldsymbol{D}$ is a finite set of variables and $\boldsymbol{V}$ is a finite set of functions. Each element of $\boldsymbol{V}$ is a pair of symbols, the name of the function and its domain.

Based on the model used, the system will return

- *true* if the FOL formula is satisfiable,
- *false* if the formula is unsatisfiable, and
- *undefined* if there is an undefined function in the formula.

Figure 2 shows outputs from model checking based on an example model and three formulas.

3 Method

3.1 System Overview

Figure 3 shows the overview of our proposed system. The system takes a table and a sentence as inputs and determines whether the table entails, contradicts, or is neutral toward the sentence. We represent the meaning of tables as FOL structures (see Section 3.2) and the meaning of sentences as FOL formulas (see Section 3.3). In the process of translating a table, we first make a filtered table, and then translate that table to an FOL structure.

In the process of translating a sentence, we convert the sentence to a CCG derivation tree using a CCG parser (Yoshikawa et al., 2017). Before parsing, we use a Named Entity Recognition (NER) system in spaCy[2] to identify a proper noun in sentences and add extra underscores to spaces and at the end of phrases so that such phrases can be categorized as one proper noun. This derivation tree is modified by a tree transformation so that it handles numerical expressions correctly. For the tree transformation, we use tsurgeon (Levy and Andrew, 2006) (see Appendix A for more details). We then construct semantic representations (FOL formulas) of the hypothesis sentences according to the CCG derivation tree. For semantic parsing based on CCG, we use ccg2lambda (Martínez-Gómez et al., 2016). As a result, we obtain an FOL formula representing the whole sentence.

We apply model checking between the FOL structure and the FOL formula for inference using NLTK with optimization (see Section 3.4). Under the FOL formula and the FOL structure, we assume

- *entailment* if our system returns *true*,
- *contradiction* if our system returns *false*, and
- *neutral* otherwise.

3.2 Meaning Representations for Tables

The top of the Figure 3 shows the processes of translating from premise tables to FOL models. We select the **Children** and **Parents** rows from the table (a) using rows filtering (see Section 3.2.1). Then, the filtered table (b) is translated into an FOL structure (c). In (c), have is a meta-predicate (see Section 3.2.2), a predicate connecting a title and other values.

3.2.1 Rows Filtering

To isolate rows from a premise table that are related to the hypothesis sentence, we apply Distracting Rows Removal (DRR), which was proposed by the previous approach (Neeraja et al., 2021). Since that approach was NN-based, a sentence vector representation was generated for each row in the table, and the original DRR was applied to the sentence representation. Then, the similarity score between each generated sentence and the hypothesis sentence was calculated. In this process, the previous approach used fastText (Joulin et al., 2016) to obtain the embedding vectors of words. They represented a hypothesis vector sequence of length p as $(\boldsymbol{h}_0, \boldsymbol{h}_1, \dots, \boldsymbol{h}_{p-1})$ and an i-th row

[2] https://github.com/explosion/spaCy

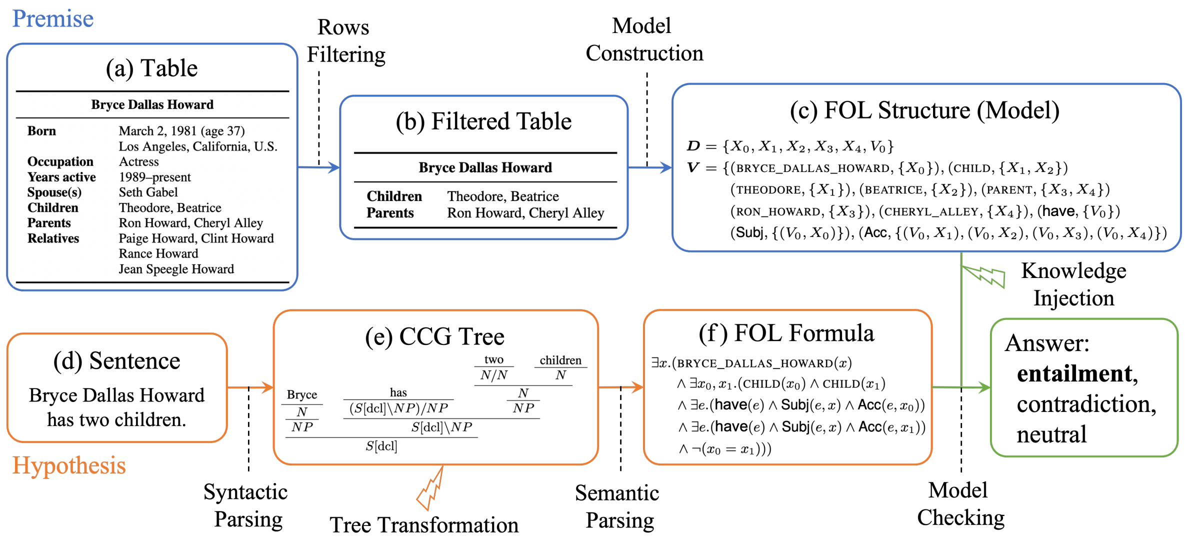

Figure 3: Overview of our proposed system with the example set for premise-hypothesis pair describing Bryce Dallas Howard. Our system returns *true* (*entailment*) for this pair.

vector sequence of length q as $(\boldsymbol{t}_0^i, \boldsymbol{t}_1^i, \ldots, \boldsymbol{t}_{q-1}^i)$. The similarity score was then calculating using

$$\text{SCORE}_i = \sum_{0 \le j < p} \max_{0 \le k < q} (\boldsymbol{h}_j \cdot \boldsymbol{t}_k^i)$$

Finally, the four rows which were the most similar were selected as the premise.

We follow most of the original DRR, but with a slight modification. First, since we directly represent a set of rows as FOL structures, we do not need to generate a sentence for each row. Thus, our system makes a simple concatenation (not using any words) of keys and values rather than a proper sentence. Also, to improve the similarity score calculation, we include numbers in a list of stopwords. In rows filtering, we select the top two most similar rows as the premise.

3.2.2 Model Construction

We construct a model based on the title and rows selected in Section 3.2.1. First, we define an entity variable X_0 that indicates a title. For keys and values in rows,

- when the key is a noun, we define entity variables X_i $(i \geq 1)$ indicating the value of each, and

- when the key is a verb, we define event variables V_j $(j \geq 1)$, whose subject is the title entity and whose accusative is the value of each.

To classify the parts of speech of the keys as nouns or verbs of the keys, we use spaCy for part-of-speech (POS) tagging. Keys are usually composed of nouns, verbs, adjectives, and prepositions, as shown in Figure 1. Since morphosyntactic ambiguity rarely appears in keys, we can classify keys into nouns and verbs by simply using a POS tagger.

We also introduce a meta-predicate have, with an event variable V_0. The subject of have is the variable X_0 indicating the title entity, and the accusatives are any of the entities in values.

3.2.3 Knowledge Injection

In some inference problems, an inference system needs to capture paraphrases (restatements of phrases that have the same meaning but are worded differently) in a premise table and a hypothesis sentence. For example, the function WIFE is injected in a model because *spouse* can be paraphrased as *wife*.

Using knowledge graphs to paraphrase some words in keys, we calculate the relatedness score between each word in keys (*key_term*) and each word in the hypothesis sentence (*hypo_term*). When the score exceeds the threshold (0.5), the *hypo_term* is introduced as a function, and the domain of which is the same as that of the *key_term*. In this process, we use the standard knowledge graph ConceptNet (Liu and Singh, 2004) to get the relatedness score between *key_term* and *hypo_term*. ConceptNet is a knowledge base that

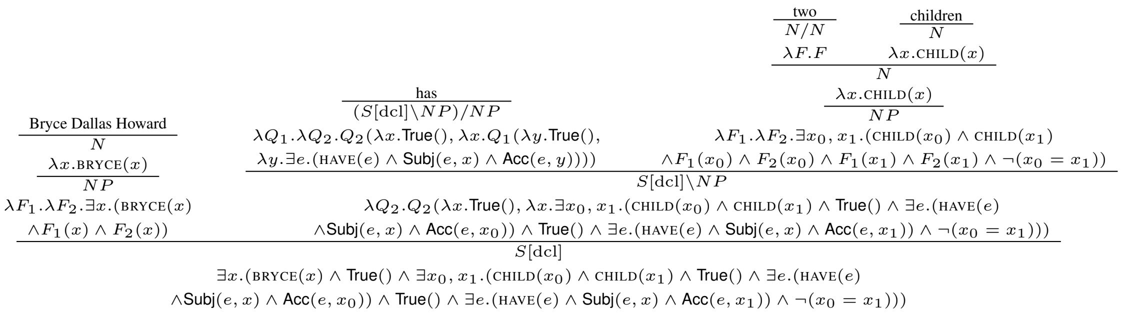

Figure 4: A derivation tree of *Bryce Dallas Howard has two children.* True is a predicate which always returns true regardless of arity and argument. The function BRYCE is an abbreviation for BRYCE_DALLAS_HOWARD_.

	Phrase	Logical formula
(a)	less than two books	$\lambda F_1 F_2.\forall x_0 x_1.((\text{BOOK}(x_0)\wedge\text{BOOK}(x_1) \wedge F_1(x_0) \wedge F_2(x_0) \wedge F_1(x_1) \wedge F_2(x_1))\rightarrow(x_0=x_1))$
(b)	at least two books	$\lambda F_1 F_2.\exists x_0 x_1.(\text{BOOK}(x_0) \wedge \text{BOOK}(x_1) \wedge F_1(x_0) \wedge F_2(x_0) \wedge F_1(x_1) \wedge F_2(x_1) \wedge \neg(x_0=x_1))$
(c)	twice	$\lambda VQK.\exists e_1 e_2.(V(Q, \lambda e.(K(e) \wedge (e = e_1))) \wedge V(Q, \lambda e.(K(e) \wedge (e = e_2))) \wedge \neg(e_1 = e_2))$

Table 1: Examples of FOL formulas. F_1 and F_2 in (a) and (b) are unary predicates representing additional attributes of *books* on the bottom of the syntactic tree. In (c), V is a unary predicate for verb phrases, Q is a binary predicate for noun phrases, and K is a unary predicate for additional attributes of the event.

includes WordNet (Miller, 1995). We select ConceptNet because InfoTabS requires paraphrases based on not only hypernymy and hyponymy relations considered in WordNet, but also common knowledge. For example, to understand whether the hypothesis *Joe Biden has married twice* is entailed or not by Figure 1, we need to capture paraphrases between **Spouse** in the premise table and *marry* or *marriage* in the hypothesis.

3.3 Meaning Representations for Sentences

We construct meaning representations of hypothesis sentences based on the CCG derivation tree and Neo-Davidsonian Event Semantics (Parsons, 1990). ccg2lambda (Mineshima et al., 2015; Martínez-Gómez et al., 2016) is used to obtain meaning representations (FOL formulas) of hypothesis sentences based on CCG and λ-calculus. We extend the semantic template that defines lexical entries and schematic entries assigned to CCG categories in Mineshima et al. (2015) so that it can handle the numerical expressions for this task. In total, we add 251 extra lexical entries for the numerical expressions. Figure 4 shows an example of CCG derivation trees with meaning representations involving numerical expressions.

We focus on expressions related to numerical comparatives: *less than*, *no more than*, *exactly*, *at least*, *no less than*, and *more than*. We need to consider how to represent the meaning of a noun phrase (NP as its CCG category) that involves a numerical comparative and the number of entities, such as *less than two books*. The meaning of this phrase is analyzed in Table 1a. We also analyze the meaning of the phrase *at least two books* in Table 1b. The meaning representation of *exactly two books* is given as the composition of the representation of *at least two books* and the representation of *no more than two books* (van Benthem, 1986).

Adverbs of frequency such as *twice* describe the number of events, and their CCG category is $(S\backslash NP)\backslash(S\backslash NP)$. The semantic representation of *twice* is given in Table 1c.

In previous work, Haruta et al. (2020a,b) handled generalized quantifiers including numerical comparatives as binary predicates many. For example, the noun phrase *two cats* is represented as $\text{CAT}(x) \wedge \mathsf{many}(x, 2)$, which indicates that x has the property of CAT and is composed of at least 2 entities. Since one of the aims of our system is to count the elements in the values of premise tables, our system assigns different entities for every word or phrase in the values.

3.4 Optimization of Model Checking

To optimize the process of model checking between tables and texts, we extend the implementation of model checking in NLTK. Figure 5 shows the program that evaluates the truth-value of $\exists x.A$. NLTK is implemented in Python and uses a set,

```
1: for y in D do
2:     if the truth-value of A[y/x] is true then
3:         return true
4:     end if
5: end for
6: return false
```

Figure 5: A program for evaluating the truth-value of $\exists x.A$.

Karachi	
Country	Pakistan
Province	Sindh
Metropolitan corporation	2011
City council	City Complex, Gulshan-e-Iqbal Town
Districts	Central Karachi, East Karachi, South Karachi, West Karachi, Korangi, Malir

Table 2: The premise table for the hypothesis *Karachi has a half dozen districts.*

which is an unordered collection, to represent the domain $\boldsymbol{D}$ of an FOL structure. When evaluating a for loop with a set (line 1 of Figure 5), an order of values in the set is not fixed for each run. To fix the order, we changed the implementation of the domain from a set to a list.

We also modify the original program for model checking in NLTK to make judgments faster. First, we sort the domain $\boldsymbol{D}$ to facilitate faster evaluation, giving $(X_0, X_1, \ldots, X_{n-1}, V_0, V_1, \ldots, V_{m-1})$, where n and m are the number of entities and events, respectively. It is sorted this way because the title variable X_0 is often the subject of the hypothesis sentence, which can be found at the top of the meaning representations.

Second, we use constraints for both the existential and universal quantifiers ($\exists$ and $\forall$). We do not substitute one variable for the other type of bounded variable in the evaluation scheme during quantification. Third, we use constraints for existential quantifiers ($\exists$) so as not to use the same variables for two or more bounded variables during substitution. We apply this restriction for only to entity variables because the same variable may be applied to different bounded variables for each event. In the process of model checking, we set a timeout of 10 seconds for judging whether the formula is satisfiable.

4 Experiments

We evaluate the extent to which our system can perform inference with numerical comparatives. We make an evaluation protocol that focuses on the numerical understanding between semi-structured tables and texts in English.

4.1 Dataset

We created a new dataset for the numerical understanding of semi-structured tables. There are two motivations for doing so. One is that the number of test cases for numerical understanding is limited to the previous NLI dataset for semi-structured tables, InfoTabS (Gupta et al., 2020). In addition, to evaluate whether NLI systems consistently perform inference with numerical comparatives, we need to analyze whether the prediction labels change correctly when the numbers in the hypothesis sentence are slightly changed from those in the original hypothesis sentence.

To create the dataset for numerical understanding of semi-structured tables, we first manually extracted 105 examples involving numerical expressions from the α_1, α_2, and α_3 test sets in InfoTabS. The inference for these examples requires an understanding of the number of entities and events. We then made a *problem set* from each example and defined the *base hypothesis* of the test cases by rewriting to the actual value n with *exactly* entailed from a premise table.

Table 2 shows a premise table for the hypothesis *Karachi has a half dozen districts*, which was extracted from InfoTabS. This premise-hypothesis pair is an example, and it makes a problem set for the statement *how many districts Karachi has*. Because we can precisely see six districts in Karachi from the premise table, the base hypothesis of this problem set is *Karachi has exactly six districts*, where *a half dozen* is defined as the number *six*. When the gold label of an example extracted from InfoTabS is *neutral*, a base hypothesis of the example is made by simply replacing the numerical comparatives with *exactly*. The gold label of the base hypothesis is the same as that of the original example. For instance, if the original hypothesis is *Bob has more than two dogs*, and its gold label is *neutral*, then the base hypothesis becomes *Bob has exactly two dogs*. Finally, we make test cases from each base hypothesis using the following process:

(i) We make a new hypothesis sentence S by removing *exactly* from the base hypothesis.

Hypothesis	Gold	Note
Karachi has less than five districts.	C	[2]
Karachi has less than six districts.	C	[1]
Karachi has less than seven districts.	E	
Karachi has five districts.	E	[1]
Karachi has six districts.	E	
Karachi has seven districts.	C	
Karachi has more than five districts.	E	[1]
Karachi has more than six districts.	C	
Karachi has more than seven districts.	C	

Table 3: A part of the test cases made from the problem set for the base hypothesis *Karachi has exactly six districts*. $[i]$ $(i = 1, 2)$ as noted means that the test case is not defined when $n \leq i$, n being the actual value. E and C are *entailment* and *contradiction*, respectively.

(ii) We make two new hypothesis sentences, S_+ and S_- by replacing the number n in S with $n + 1$ and $n - 1$ in S, respectively.

(iii) We make six additional hypothesis sentences each from S, S_+, and S_- by adding the expressions related to numerical comparatives, *less than*, *no more than*, *exactly*, *at least*, *no less than*, and *more than*, thus making a problem set consisting of 21 hypothesis sentences with correct gold labels. Table 3 shows a part of the hypothesis sentences.

(iv) We remove unnatural hypothesis sentences from the problem set, including such as *at least zero* and *less than one*.

Note that here *two* has the same meaning as *at least two*. Our dataset consists of 105 problem sets with 1,979 test cases. The distribution of gold labels is (*entailment*, *neutral*, *contradiction*) = (965, 176, 838). This dataset includes ten problem sets that are filled with *neutral* labels. We confirmed all words are commonly used in a training set in InfoTabS and our dataset.

4.2 Experimental Setup for Previous Research

Neeraja et al. (2021) proposed an NN-based model for inference between semi-structured tables and texts and tested it by InfoTabS. We compare our system to +KG explicit, which was the setting for which the previous model (Neeraja et al., 2021) achieved the highest performance. +KG explicit consists of the following four methods for making sentence representations of tables.

	+KG	Ours
All problem sets	0.03	**0.31**
All problem sets excluding *neutral*-filled	0.00	**0.27**

Table 4: The accuracy of problem sets whose test cases were all predicted correctly. +KG is an abbreviation for +KG explicit.

Implicit Knowledge Addition The model adds information that is not in the tables and texts to models by pre-training with a large-scale NLI corpus, MultiNLI (Williams et al., 2018).

Better Paragraph Representation The model generates more grammatical sentences for specific entity types, such as money, date, and cardinal, with carefully crafted templates when making sentence representations of tables.

Distracting Rows Removal (DRR) The model removes several rows from the premise table that are unrelated to the hypothesis sentence. For a detailed explanation of DRR, see Section 3.2.1.

Explicit Knowledge Addition The model adds a suitable meaning to the keys for each premise from WordNet (Miller, 1995) or Wikipedia articles by calculating similarity based on the BERT embedding.

+KG explicit makes sentence representations of tables and uses RoBERTa-large (Liu et al., 2019) for encoding premise-hypothesis pairs. Almost all of the setups are identical to what was used in previous research except (i) the batch size is set to 4 and (ii) we adopt the result of one seed rather than the average of three seeds.

4.3 Results

Accuracy per Problem Set Table 4 shows the accuracy of the previous model (+KG) and our system (Ours) for a number of problem sets. Our proposed system could correctly predict 31% of all problem sets, while the previous model only predicted 3%. Premise-hypothesis pairs whose gold labels are *neutral* can be predicted correctly without a precise numerical understanding. Table 4 also shows that +KG could not perform inference on any problem set whose gold labels were *entailment* or *contradiction* at all. On the other hand, the accuracy of our logic-based system was 27%. These results indicate that our system better handles inference involving numerical comparatives

	+KG	Ours
less than k	0.10	**0.36**
no more than k	0.10	**0.35**
exactly k	0.19	**0.32**
k	0.24	**0.33**
at least k	0.08	**0.32**
no less than k	0.19	**0.33**
more than k	0.17	**0.35**

Table 5: The accuracy for each numerical comparative construction. +KG is an abbreviation for +KG explicit. k indicates a number.

than the previous model, being able to more robustly predict *entailment* and *contradiction* labels. This shows that our proposed dataset for numerical understanding is challenging for current systems. We describe the error analysis of our system in the fourth paragraph of this section.

Understanding for Each Numerical Comparative Table 5 shows the accuracy of both methods for each numerical comparative construction. We observe that our proposed method can predict correct labels more often than the existing method for all numerical comparatives.

Run Time for Model Checking with Optimization We compare the run times for model checking with and without our optimization for model checking (see Section 3.4). We chose six problem sets involving different numbers of values, which consist of two problem sets each whose numbers of values are 2, 4, and 6. All of the problems require understanding the number of entities. The number of test cases is 124. Table 6 shows the average and maximum run times for ten trials. We observe that our optimization made model checking much faster.

Error Analysis Error analysis shows that main errors are caused by the failure of knowledge injection. Figure 6 shows two premise-hypothesis pairs, one for which our system was able to perform inference and one for which it was not. In Figure 6a, the function HUSBAND was added to the model in the knowledge injection process because the relatedness score between *spouse* and *husband* was high (0.747). On the other hand, in Figure 6b, the function WIN was not added to the model because the relatedness score between *award* and *win* was low (0.336). In addition, even though we improved the speed of the original model checking program, several test cases still ran out of time.

Optimization	Average	Maximum
disabled	3.20	185.17
enabled	**0.04**	**1.26**

Table 6: Average and maximum run time (seconds) for model checking with and without optimization.

For example, the problem with the hypothesis sentence *Jimmy Eat World has been on 13 labels* (this gold label is *contradiction*) exceeded the maximum time limit (10 seconds).

Discussion We discuss how to handle various types of inference other than the numerical one in InfoTabS with our inference system. First, we have to correctly parse values in various tables and extract information from them. For example, to determine whether Hypothesis 1 in Figure 1 is entailed by the premise table, we need to parse the noun phrase *November 20, 1942* into one date format. In addition to this, various formats are needed to be provided, such as age, duration, and year of marriage. Also, some test cases require arithmetic operations other than counting, such as *Joe Biden and Neilia Hunter divorced six years after their marriage*, based on the premise table in Figure 1. Although such issues are tricky, we believe that our logic-based approach is applicable with adding premises related to arithmetic operations.

5 Conclusion

In this study, we proposed a logic-based system for an NLI task that requires numerical understanding in semi-structured tables. We built an NLI dataset that focuses on numerical comparatives between semi-structured tables and texts. Using this dataset, we showed that our system performed more robustly than the previous NN-based model.

In future work, we will improve knowledge injection process to cover various problems. We also seek to handle other generalized quantifiers such as *many*. We believe that our system and dataset for performing numerical inference between semi-structured tables and texts could pave the way for applications of inference between resources other than texts.

Acknowledgements

We thank the three anonymous reviewers for their helpful comments and feedback. This work was

Jodie Whittaker	
Spouse	Christian Contreras

i. Part of the filtered table describing Jodie Whittaker.

$$\begin{aligned} \boldsymbol{D} &= \{X_0, X_1, V_0\} \\ \boldsymbol{V} &= \{(\textsc{jodie\_whittaker\_}, \{X_0\}), (\textsc{spouse}, \{X_1\}), \\ &\quad (\textsc{christian\_contrerasm}, \{X_1\}), \\ &\quad \underline{(\textsc{husband}, \{X_1\})}, (\textsc{have}, \{V_0\}), \\ &\quad (\mathsf{Subj}, \{(V_0, X_0)\}), (\mathsf{Acc}, \{(V_0, X_1)\})\} \end{aligned}$$

ii. Part of the model constructed by our system for (a-i).

$$\exists x.(\textsc{jodie\_whittaker\_}(x) \wedge \exists x_0.(\underline{\textsc{husband}(x_0)} \wedge \exists e.(\mathsf{have}(e) \wedge \mathsf{Subj}(e, x) \wedge \mathsf{Acc}(e, x_0))))$$

iii. An FOL formula constructed from the hypothesis *Jodie Whittaker has had one husband.*

(a) Outputs of our system to the premise-hypothesis pair describing Jodie Whittaker. Our system was able to perform inference correctly.

Karl Ferdinand Braun	
Awards	Nobel Prize in Physics (1909)

i. Part of the filtered table describing Karl Ferdinand Braun.

$$\begin{aligned} \boldsymbol{D} &= \{X_0, X_1, V_0\} \\ \boldsymbol{V} &= \{(\textsc{karl}, \{X_0\}), (\textsc{award}, \{X_1\}), \\ &\quad (\textsc{nobel\_prize\_physic}, \{X_1\}), (\textsc{have}, \{V_0\}), \\ &\quad (\mathsf{Subj}, \{(V_0, X_0)\}), (\mathsf{Acc}, \{(V_0, X_1)\})\} \end{aligned}$$

ii. Part of the model constructed by our system for (b-i).

$$\exists x.(\textsc{karl}(x) \wedge \exists x_0.(\textsc{award}(x_0) \wedge \exists e.(\underline{\textsc{win}(e)} \wedge \mathsf{Subj}(e, x) \wedge \mathsf{Acc}(e, x_0))))$$

iii. An FOL formula constructed from the hypothesis *Karl Ferdinand Braun won one award.*

(b) Outputs of our system to the premise-hypothesis pair describing Karl Ferdinand Braun. Our system was not able to perform inference correctly.

Figure 6: Two premise-hypothesis pairs, one for which our system was able to perform inference (a) and one for which it was not (b). The function KARL in (b-ii, b-iii) is an abbreviation for KARL_FERDINAND_BRAUN_. The underlined functions are added in the knowledge injection process to perform inference.

supported by PRESTO, JST Grant Number JPMJPR21C8, Japan.

References

Lasha Abzianidze. 2015. A tableau prover for natural logic and language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2492–2502, Lisbon, Portugal. Association for Computational Linguistics.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Johan Bos. 2008. Wide-coverage semantic analysis with Boxer. In *Semantics in Text Processing. STEP 2008 Conference Proceedings*, pages 277–286. College Publications.

Stephen Clark and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

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, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dan Garrette and Ewan Klein. 2009. An extensible toolkit for computational semantics. In *Proceedings of the Eight International Conference on Computational Semantics*, pages 116–127, Tilburg, The Netherlands. Association for Computational Linguistics.

Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. 2020. INFOTABS: Inference on tables as semi-structured data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2309–2324, Online. Association for Computational Linguistics.

Izumi Haruta, Koji Mineshima, and Daisuke Bekki. 2020a. Combining event semantics and degree semantics for natural language inference. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1758–1764, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Izumi Haruta, Koji Mineshima, and Daisuke Bekki. 2020b. Logical inferences with comparatives and generalized quantifiers. In *Proceedings of the*

58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pages 263–270, Online. Association for Computational Linguistics.

Hai Hu, Qi Chen, Kyle Richardson, Atreyee Mukherjee, Lawrence S. Moss, and Sandra Kuebler. 2020. MonaLog: a lightweight system for natural language inference based on monotonicity. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 334–344, New York, New York. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. FastText.zip: Compressing text classification models. *Computing Research Repository*, arXiv:1612.03651.

Jayant Krishnamurthy, Pradeep Dasigi, and Matt Gardner. 2017. Neural semantic parsing with type constraints for semi-structured tables. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1516–1526, Copenhagen, Denmark. Association for Computational Linguistics.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

Roger Levy and Galen Andrew. 2006. Tregex and tsurgeon: tools for querying and manipulating tree data structures. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Hugo Liu and Push Singh. 2004. Conceptnet - a practical commonsense reasoning tool-kit. *BT Technology Journal*, 22:211–226.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *Computing Research Repository*, arXiv:1907.11692.

Pascual Martínez-Gómez, Koji Mineshima, Yusuke Miyao, and Daisuke Bekki. 2016. ccg2lambda: A compositional semantics system. In *Proceedings of ACL-2016 System Demonstrations*, pages 85–90, Berlin, Germany. Association for Computational Linguistics.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Koji Mineshima, Pascual Martínez-Gómez, Yusuke Miyao, and Daisuke Bekki. 2015. Higher-order logical inference with compositional semantics. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2055–2061, Lisbon, Portugal. Association for Computational Linguistics.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

J. Neeraja, Vivek Gupta, and Vivek Srikumar. 2021. Incorporating external knowledge to enhance tabular reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2799–2809, Online. Association for Computational Linguistics.

Terence Parsons. 1990. *Events in the Semantics of English: A Study in Subatomic Semantics*. The MIT Press, Cambridge, MA.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China. Association for Computational Linguistics.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361, Hong Kong, China. Association for Computational Linguistics.

Kyle Richardson, Hai Hu, Lawrence Moss, and Ashish Sabharwal. 2020. Probing natural language inference models through semantic fragments. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8713–8721.

Ohad Rozen, Vered Shwartz, Roee Aharoni, and Ido Dagan. 2019. Diversify your datasets: Analyzing generalization via controlled variance in adversarial datasets. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 196–205, Hong Kong, China. Association for Computational Linguistics.

Mark Steedman. 2000. *The Syntactic Process*. The MIT Press, Cambridge, MA.

Riko Suzuki, Hitomi Yanaka, Masashi Yoshikawa, Koji Mineshima, and Daisuke Bekki. 2019. Multimodal logical inference system for visual-textual entailment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 386–392, Florence, Italy. Association for Computational Linguistics.

Johan van Benthem. 1986. *Essays in Logical Semantics*. Springer, Dordrecht.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Masashi Yoshikawa, Hiroshi Noji, and Yuji Matsumoto. 2017. A* CCG parsing with a supertag and dependency factored model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 277–287, Vancouver, Canada. Association for Computational Linguistics.

Knowledge injection	Accuracy
disabled	0.23
enabled	**0.34**

Table 7: The accuracy of our proposed system with and without knowledge injection.

A Examples of Tree Transformation

We detect where to transform by tregex (Levy and Andrew, 2006), the regular expression for trees. We have three tsurgeon scripts, all of which are for handling numerical expressions involving the number of events. For example, as Figure 7 shows, we transform the CCG subtree (a) for *exactly* n *times*, where n is a number, into the CCG subtree (b).

B Ablation Study for Knowledge Injection

We conducted an ablation study for knowledge injection (see Section 3.2.3). We picked all of the base hypotheses in our dataset (105 cases in total) and experimented to see how effective our knowledge injection method is. As seen in Table 7, our knowledge injection method provided increased accuracy by 11% (12 cases).

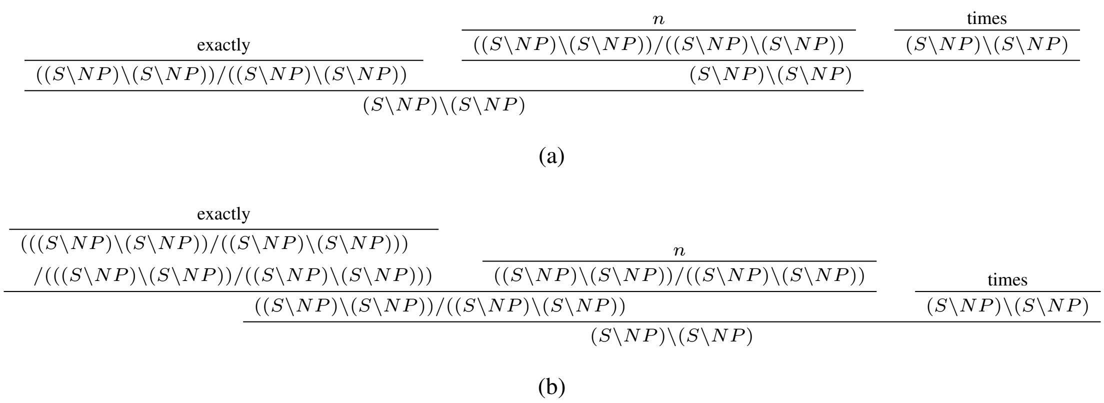

Figure 7: An example tree transformation process for *exactly* n *times*, where n is a number. (a) is transformed into (b).