# Natural Language Inference Using Neural Network and Tableau Method

**Ayahito Saji[1], Daiki Takao[1], Yoshihide Kato[2], Shigeki Matsubara[1,2]**
[1]Graduate School of Informatics, Nagoya University
[2]Information and Communications, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-8601, Japan
saji.ayahito.y7@s.mail.nagoya-u.ac.jp

## Abstract

Natural language inference (NLI) is the task of identifying the inferential relation between a text pair. Although neural-based approaches have recently achieved high performance on the NLI task, they are unable to explain their reasoning processes. Symbolic approaches, on the other hand, have the advantage that the reasoning process is understandable to humans. This paper proposes a method for integrating a neural NLI model and tableau proof system, with the latter explaining the reasoning processes. The standard tableau method consists of decomposing logical formulas by applying inferential rules, and checking whether or not there exists a valuation that satisfies the given constraints. Unlike the standard tableau method, our method uses dependency structures as its ingredients rather than logical formulas, and it employs a neural NLI model for the latter process. To analyze the behavior of a neural NLI model, we conducted an experiment on the neural NLI model alone and proposed method using the SNLI corpus.

## 1 Introduction

Natural language inference (NLI) is the task of identifying the inferential relation between a text pair: *premise* and *hypothesis*. If a hypothesis can be inferred from a premise using logical and common sense knowledge, it is judged as *entailment*; if they are incompatible, it is judged as *contradiction*; and if neither of them is the case, it is judged as *neutral*. It is expected to be used in a wide range of fields such as question answering, information retrieval, and text summarization.

In recent years, neural-based approaches have achieved high performance on the NLI task. For example, Chen et al. (2017) proposed a model based on word embedding and bidirectional LSTM (Hochreiter and Schmidhuber, 1997). Dispite its simplicity, the model achieves a high accuracy in experiments with the SNLI corpus (Bowman et al., 2015). On the other hand, neural-based approaches have the limitation that the models cannot explain the reasoning processes that lead to judgment. The model is a black box, and it is difficult to analyze what kind of inference was performed. Furthermore, Gururangan et al. (2018) demonstrated that NLI datasets such as the SNLI corpus and MultiNLI corpus (Williams et al., 2018) have a hidden bias in that inferential relations can be determined only from a hypothesis, and they highlighted the risk that neural models are simply identifying inferential relations based on the biases.

On the other hand, symbolic approaches to the NLI task have been proposed. These approaches have the advantage that the reasoning process leading to the result is understandable to humans, unlike the neural model approach. In addition, symbolic manipulations in these approaches are generally founded on formal logic or linguistic analyses, which allow us to understand the reasoning processes.

In this paper, we propose a method to add to the neural NLI model the ability of the symbolic manipulation approach that makes the reasoning process explicit. Our approach only assumes that an

NLI model takes a pair of premises and a hypothesis and outputs inferential relation. That is, our approach can be applied to any neural NLI model. Our method combines a neural NLI model and a tableau proof system. The standard tableau method consists of decomposing logical formulas by applying inference rules, and checking whether or not there exists a valuation that satisfies the given constraints. For the latter process, our method uses a neural NLI model. Unlike the standard tableau method, our approach uses dependency structures as its ingredients, rather than logical formulas. This characteristic enables us to integrate a neural NLI model into a symbolic proof system.

## 2 Tableau Method

In this section, we describe the standard tableau method, which is the basis of our proposed method. The tableau method is a procedure for proving whether, given a set of pairs of logical formulas and truth values (called *entries* in the following), there exists a valuation that assigns a truth value to each logical formula in the set. The relation between NLI and the tableau method can be summarized as follows:

- A premise and hypothesis are in a contradiction relation when the procedure proves that there is no valuation such that both the premise and hypothesis are true.

- When the procedure demonstrates that there is no valuation where the premise is true and the hypothesis is false, it implies that if the premise is true, then the hypothesis is also true, i.e., the premise implies the hypothesis.

- If neither can be proved, it means that the neutral relation holds between the premise and hypothesis.

The tableau method constructs a tree structure, called a *tableau*, for a given set of entries $E$. Each node in the tableau is labeled with an entry $[X : A]$. This entry represents the constraint that the logical formula $A$ must take $X$ as its truth value. The initial tableau is made up of nodes labeled with $E$ elements. The tableau is created by applying the tableau rules to the nodes repeatedly. The constraints expressed by the entries are decomposed into constraints on subformulas using the tableau rules. The decomposed constraints are then added to the tableau as new nodes. Branches in tableau mean that there are multiple cases for possible valuation.

When nodes on the path from the root to a leaf of a tableau have the labels $[\mathrm{T} : A]$ and $[\mathrm{F} : A]$ (where T and F denote true and false, respectively), we say that this path is *closed*. If all paths in a tableau are closed, we say that the tableau is closed. The fact that the tableau is closed means that no valuation satisfies the constraints represented by $E$.

## 3 Proposed Method

In this section, we propose an NLI system that combines a tableau method based on dependency structures and neural model-based judgment of closed tableaux. Our system performs the following steps:

**Dependency parsing.** Convert the premise and hypothesis texts into dependency structures $D_P$ and $D_H$, respectively.

**Inference based on the tableau method.** For the dependency structures $D_P$ and $D_H$, our system constructs two tableaux. One proves entailment relation (a tableau derived from $[\mathrm{T} : D_P], [\mathrm{F} : D_H]$) and the other proves contradiction relation (a tableau derived from $[\mathrm{T} : D_P], [\mathrm{T} : D_H]$). In the following, we refer to these tableaux as *entailment tableau* and *contradiction tableau*, respectively.

**Checking closed tableau.** Determine whether the entailment and contradiction tableaux are closed using a neural NLI model.

### 3.1 Dependency parsing

Our proposed method uses a neural NLI model to determine the closed tableau, and the model takes natural language sentences as inputs. To accomplish this, we use dependency structure as an entry in the tableau rather than logical formulas. The premise and hypothesis sentences are converted into dependency structures in this step. As a dependency formalism, We adopt Universal Dependencies (UD) (McDonald et al., 2013). For this step, we can use any UD-based dependency parser. Figure 1 depicts a dependency structure.

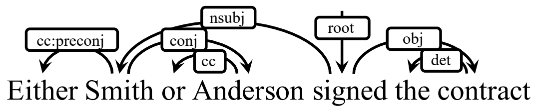


Figure 1: The dependency structure of “Either Smith or Anderson signed the contract”

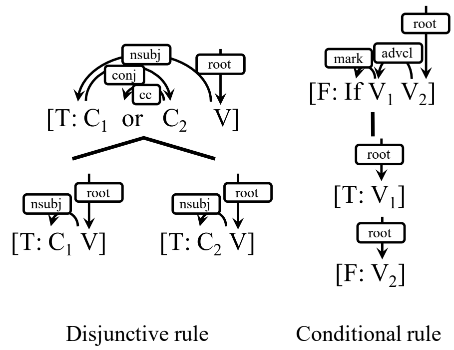


Figure 2: Examples of tableau rules

### 3.2 Inference based on the tableau method

In this section, we explain the tableau method based on the dependency structure.

#### 3.2.1 Tableau rules

Each tableau is derived by applying *tableau rules* as in the standard method. All tableau rules in the proposed method are in the following form:

$$P \to (C_{1,1} \wedge \cdots \wedge C_{1,n_1}) \vee \cdots \vee (C_{m,1} \wedge \cdots \wedge C_{m,n_m}),$$

where $P, C_{1,1}, \ldots, C_{1,n_1}, \ldots, C_{m,1}, \ldots, C_{m,n_m}$ are pairs of truth values and dependency structure patterns. The dependency structure patterns contain variables, and each variable is bound to a matched dependency structure or matched dependency structure sequence. Examples of tableau rules are shown in Figure 2. If $P$ matches a node $N$ on a path of the tableau, the procedure adds new $m$ branches $\langle \sigma(C_{1,1}) \ldots \sigma(C_{1,n_1}) \rangle$, …, $\langle \sigma(C_{m,1}) \ldots \sigma(C_{m,n_m}) \rangle$ as children of the leaves of the path. Here, $\sigma$ is the function that substitutes the variables with the bounded elements.

The tableau rules decompose the constraints expressed by the source node. For example, applying the rule on the left of Figure 2 to the node with the label [T: Either Smith or Anderson signed the contract.] will add two newly nodes to the path's end (the tableau leaf). The added nodes are labeled with [T: Smith signed the contract.] and [T: Anderson signed the contract.]. The constraints expressed by the two newly added nodes are equivalent to the constraints expressed by the original nodes. There is no need to apply a new operation to nodes to which the tableau rule has been applied, because the constraint is already expressed by the node from which it was derived. Therefore, it is not necessary to handle the original node anymore. Our proposed method sets a flag for each node to distinguish whether the tableau rule is applied or not. The flagged node is not used for any further operations (application of the rule and judgment of the closed tableau). Figure 3 shows the entailment tableau for the premise “Either Smith or Anderson signed the contract.” and the hypothesis “If Smith didn't sign the contract, Anderson made an agreement.”.

### 3.3 Judgment of closed tableau

In standard tableau methods, a closed path is defined by the existence of entries on the path that differ only in their truth values. In addition, our proposed method introduces another type of definition of a closed path, which is based on a neural NLI model. An NLI model takes a premise $P$ and hypothesis $H$ as inputs and returns one of the following classes: entailment, neutral, or contradiction. In the following, we write $\mathrm{Rel}_M(P, H)$ for the class determined by the model $M$. When two nodes are labeled with $[X_1 : D_1]$ and $[X_2 : D_2]$, the following two situations are those in which it is not possible to assign a truth value.

- $X_1 = \mathrm{T}$
  $\wedge\, X_2 = \mathrm{T}$
  $\wedge\, \mathrm{Rel}_M(\mathrm{sen}(D_1), \mathrm{sen}(D_2)) = \text{contradiction}$
- $X_1 = \mathrm{T}$
  $\wedge\, X_2 = \mathrm{F}$
  $\wedge\, \mathrm{Rel}_M(\mathrm{sen}(D_1), \mathrm{sen}(D_2)) = \text{entailment}.$

Here, $\mathrm{sen}(D)$ is the sentence corresponding to the dependency structure $D$. In the proposed method, we define a path to be closed when there are two nodes on the path that satisfy either of the two conditions.

For example, the tableau in Figure 3 has a path ($\boxed{1}$ $\boxed{2}$ $\boxed{3}$ $\boxed{4}$ $\boxed{5}$ $\boxed{6}$) that contains $\boxed{6}$[T: Smith

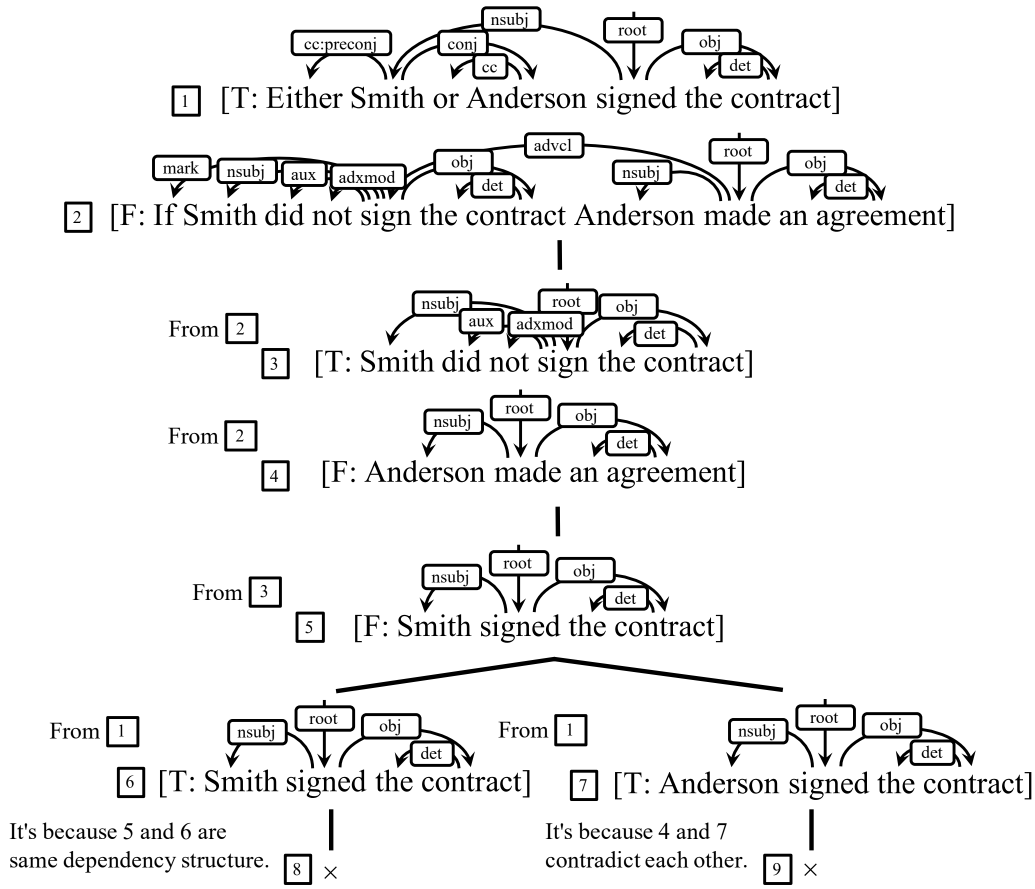


Figure 3: Example of entailment tableau

signed the contract contract] and $\boxed{5}$[F: Smith signed the contract]. Because the entries $\boxed{5}$ and $\boxed{6}$ differ only in their truth values, this path is closed in the sense of the standard tableau method. On the other hand, the other path ($\boxed{1}$ $\boxed{2}$ $\boxed{3}$ $\boxed{4}$ $\boxed{5}$ $\boxed{7}$) contains $\boxed{7}$[T: Anderson signed the contract] and $\boxed{4}$[F: Anderson made an agreement]. Assuming that $\mathrm{Rel}_M(\mathrm{sen}(\boxed{7}), \mathrm{sen}(\boxed{4})) =$ entailment, this path is closed because of our new definition.

Only those nodes that have not been given the flag described in the previous section need to be considered in determining the closed tableau.

# 4 Experiment

To analyze the behavior of a neural NLI model from the viewpoint of making the reasoning process explicit, we experimented.[1]

## 4.1 Dataset

We used the SNLI corpus (Bowman et al., 2015) as the dataset. We used the standard data split, that is, 549,367 samples for training data, 9842 for development data, and 9824 for test data.[2]

## 4.2 Dependency parsing

We used Udify (Kondratyuk and Straka, 2019), which is a multilingual dependency parser using BERT (Devlin et al., 2019). It outputs the UD-based dependency structures.

[1]The code is available at https://github.com/ayahito-saji/nli-tableau-ml.

[2]All samples classified as "unlabeled" were removed.

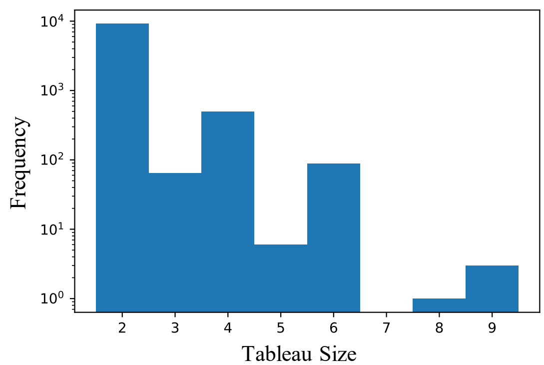


Figure 4: Distribution of entailment tableau size

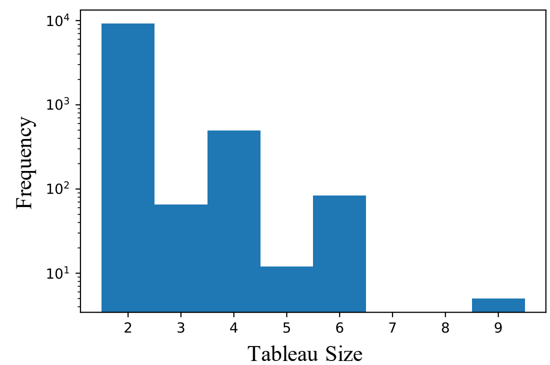


Figure 5: Distribution of contradiction tableau size

### 4.3 Neural NLI model

We used ESIM (Chen et al., 2017), which is based on LSTM.[3] The parameters of ESIM were estimated using the training and development data.

### 4.4 Tableau rules

We have created 32 tableau rules that correspond to the rules of the standard tableau method for propositional logic. There are four types of rules: conjunction, disjunction, negation, and conditional. The rules for conjunction and disjunction can handle not only coordinated sentence structure but also core arguments (subject, object, etc.) in UD. Appendix A contains the tableau rules.

### 4.5 Evaluation

The distributions of the derived tableau sizes (defined as the number of entries) are shown in Figures 4 and 5. Our tableau system could decompose the sentences for 660 of 9824 test samples.

We used the standard metrics for performance evaluation. One notable point is that, when both the entailment and contradiction tableaux were closed, they were classified as errors. The $F_1$ value is the harmonic mean of the recall and precision defined below:

$$\mathrm{Recall}_A = \frac{\mathrm{TP}_A}{\mathrm{True}_A}$$

$$\mathrm{Precision}_A = \frac{\mathrm{TP}_A}{\mathrm{Positive}_A}$$

[3]https://github.com/coetaur0/ESIM

| | | Predict | | | |
|---|---|---|---|---|---|
| | | Entail | Neut | Cont | Err |
| Ans | Entail | 138 | 21 | 25 | 38 |
| | Neut | 12 | 122 | 76 | 3 |
| | Cont | 10 | 17 | 193 | 5 |
| Accuracy | | 82.73% | 73.94% | 73.64% | - |
| Recall | | 62.16% | 57.28% | 85.78% | - |
| Precision | | 86.25% | 76.25% | 65.65% | - |
| $F_1$ | | 72.25% | 65.42% | 74.37% | - |

Table 1: Prediction results of the proposed method

Here, $\mathrm{TP}_A$ is the number of samples in which the correct answer and prediction are class $A$, $\mathrm{True}_A$ is the number of samples in which the correct answer is class $A$, and $\mathrm{Positive}_A$ is the number of samples in which the prediction is class $A$.

Tables 1 and 2 show the accuracy, recall, precision, and $F_1$ value for each class for the 660 samples where some tableau rules ware applied. The microaccuracy of the proposed method was 68.64%, the percentage classified as an error class was 6.97%, and the microaccuracy of the neural NLI model was 86.82%.

The proposed method tends erroneously to predict contradiction relation in comparison with the method using only neural models. In the next section, we analyze the reason for this.

### 4.6 Error analysis

To investigate why the proposed method failed to identify the inferential relations, we checked the tableaux constructed for such samples. In the fol-

| | | Predict | | | |
|---|---|---|---|---|---|
| | | Entail | Neut | Cont | Err |
| Ans | Entail | 195 | 20 | 7 | 0 |
| | Neut | 16 | 182 | 15 | 0 |
| | Cont | 6 | 23 | 196 | 0 |
| Accuracy | | 92.58% | 88.79% | 92.27% | - |
| Recall | | 87.84% | 85.45% | 87.11% | - |
| Precision | | 89.86% | 80.89% | 89.91% | - |
| $F_1$ | | 88.84% | 83.11% | 88.49% | - |

Table 2: Prediction results of ESIM

lowing, we will use the following actual samples:

**Premise**: Four people and a baby are crossing the street at a crosswalk.

**Hypothesis**: People and a baby crossing the street.

**Relation**: Entailment

The contradiction tableau for this sample is shown in Figure 6. Because the correct label of this sample is entailment, the tableau should not be closed. However, the tableau is closed because $\mathrm{Rel}_{\mathrm{ESIM}}(\mathrm{sen}(\boxed{3}), \mathrm{sen}(\boxed{6})) = \text{contradiction}$.

As pointed out in (Bowman et al., 2015), there is a certain indeterminacy in the inferential relation between sen($\boxed{3}$) and sen($\boxed{6}$). The inferential relation may depend on whether the entities that the noun phrases refer to are identical or not. The inferential relation between sen($\boxed{3}$) and sen($\boxed{6}$) is "contradiction" if "Four people" and "A baby" refer to the same entity, but "neutral" if they do not refer to the same entity. In the development dataset, of the 38 samples where the contradiction tableau should not be closed, the contradiction tableaux of 21 samples were incorrectly closed for the same reason. The important point to enphasize here is that we were able to capture this fact using the tableaux created by our method.

## 5 Related Work

Our proposed method adopts an approach inspired by Natural Logic (Lakoff, 1970) that performs inferences based on syntactic structures. This section gives an overview of previous Natural Logic-based methods and compares them with ours.

Natural Logic-based systems can be classified into the following two types:

- Transition-based method;
- Proof-based method.

In the transition-based approach, a premise is converted into a hypothesis using some operations. Bar-Haim et al. (2007) proposed entailment-preserving rules that transform dependency structures. NatLog is a system proposed by MacCartney and Manning (2008) that is based on an extended version of monotonicity calculus (MacCartney and Manning, 2009). In terms of directly handling natural language sentences, these methods are similar to ours. However, it is unclear how to incorporate neural NLI models into them. Furthermore, transition-based methods cannot handle multipremize problems, unlike ours.

Abzianidze (2017) proposed a tableau-proof system called LangPro. Hu et al. (2020) developed a proof system called MonaLog, which is based on monotonicity reasoning. These systems use semantic representations similar but not identical to natural language sentences. That is, it is impossible to integrate a neural NLI model with their proof system, unlike ours.

## 6 Conclusion

In this paper, we proposed an NLI system using the tableau method and a neural model. In this reported experiment, we trained the neural model using the premises and hypotheses of the dataset as they were. On the other hand, as another type of training, we will investigate how to use decomposed premises and hypotheses as training examples according to the proposed tableau method. In addition, in future research we will enrich the tableau rules and support multiple languages.

## References


Lasha Abzianidze. 2017. LangPro: Natural language theorem prover. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 115–120, Copenhagen, Denmark, September. Association for Computational Linguistics.

Roy Bar-Haim, Ido Dagan, Iddo Greental, and Eyal Shnarch. 2007. Semantic inference at the lexical-syntactic level. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pages 871–876.

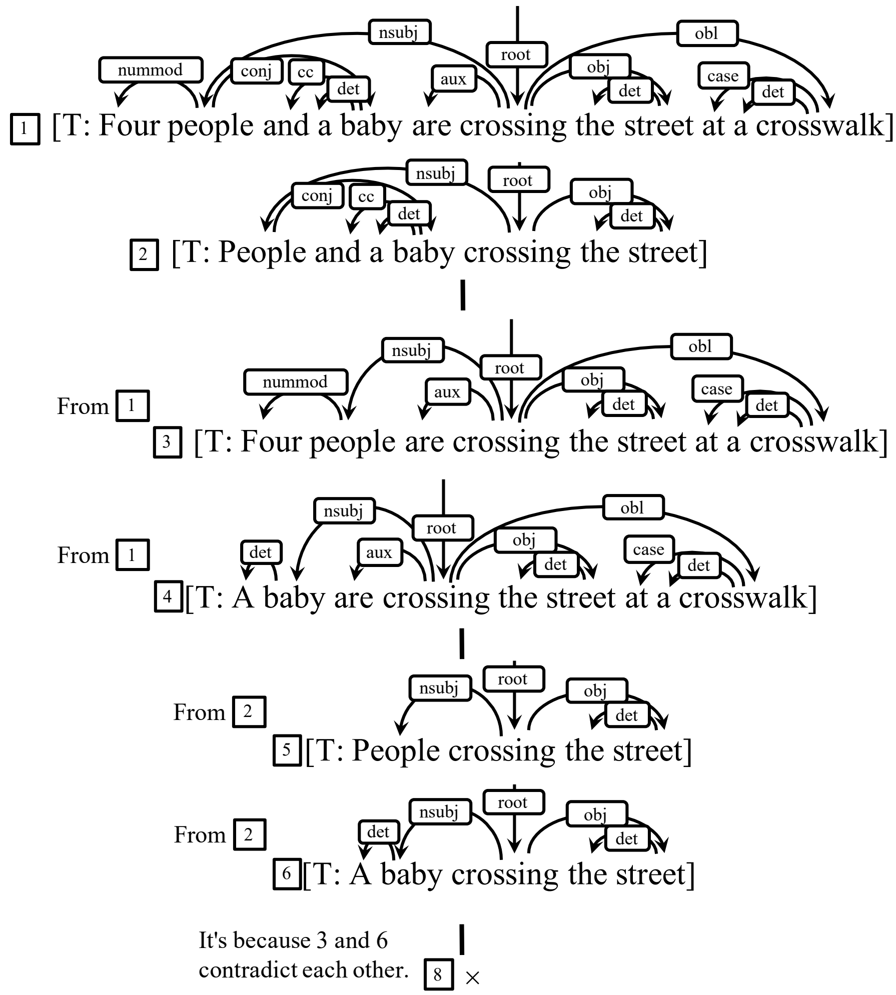


Figure 6: Example of incorrect closed tableau

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, Vancouver, Canada, July. Association for Computational Linguistics.

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, June. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana, June. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

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, January. Association for Computational Linguistics.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing Universal Dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China, November. Association for Computational Linguistics.

George Lakoff. 1970. Linguistics and natural logic. *Synthese*, 22(1):151–271.

Bill MacCartney and Christopher D. Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 521–528, Manchester, UK, August. Coling 2008 Organizing Committee.

Bill MacCartney and Christopher D. Manning. 2009. An extended model of natural logic. In *Proceedings of the 8th International Conference on Computational Semantics*, pages 140–156, Tilburg, The Netherlands, January. Association for Computational Linguistics.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal Dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria, August. Association for Computational Linguistics.

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, June. Association for Computational Linguistics.

## Appendix A

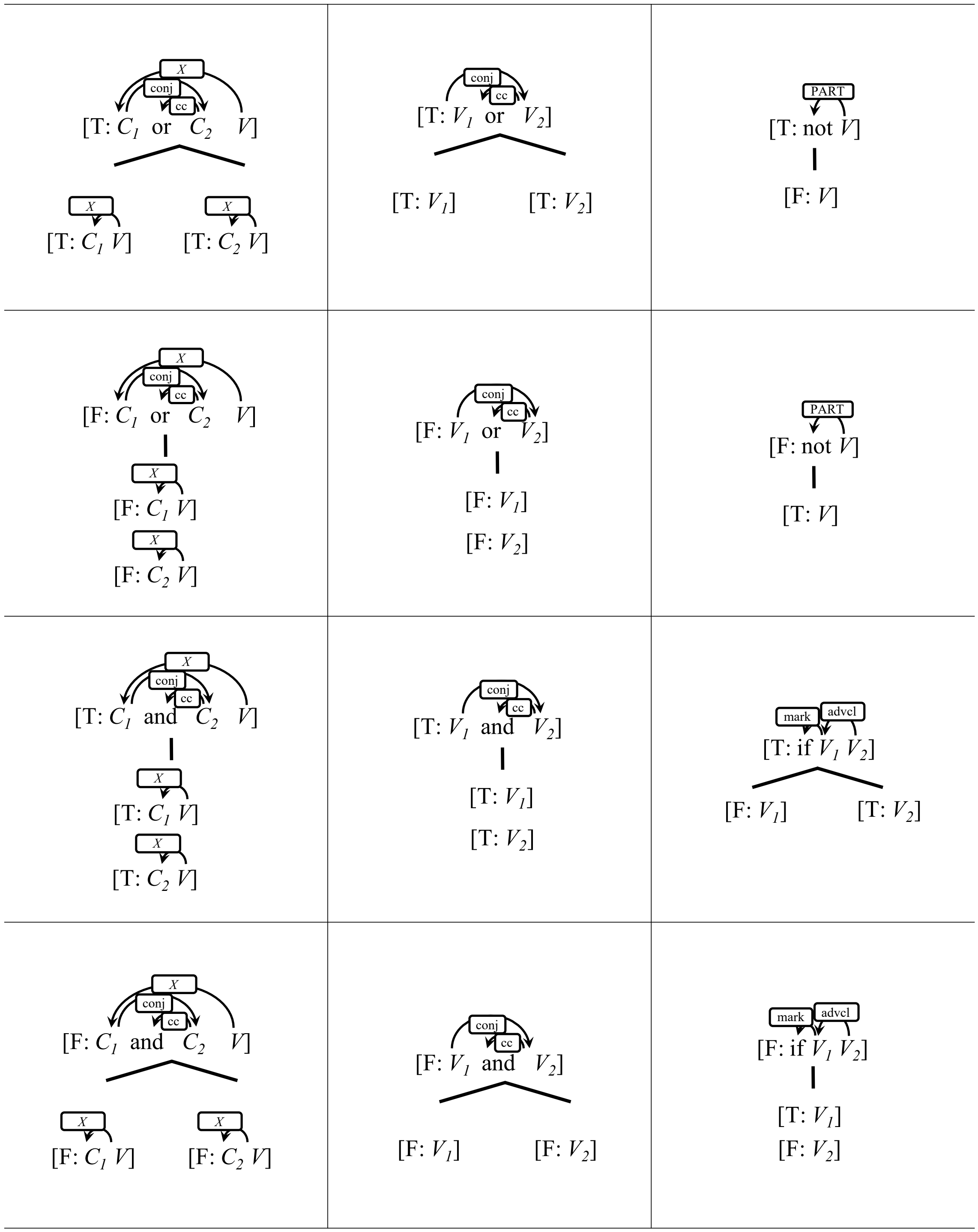


Figure 7: All rules that we created

Figure 7 shows all rules that we created. Here, $V$ is a variable matching any dependency structure whose headword is a verb, $C_i$ is a variable matching any dependency structure, and, $X$ is a dependency relation in $\{$nsubj, csubj, obj, iobj, ccomp, xcomp$\}$.