A Model-Theoretic Formalization of Natural Language Inference Using Neural Network and Tableau Method

Ayahito Saji[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

Saji et al. (2021) integrated a neural-based NLI model and a symbolic one into a framework to get the best of both worlds. This framework is based on a tableau method, which is a proof system for formal logic; however, it has a remaining issue that its theoretical limitations have not been clarified. To solve this issue, this paper formalizes the framework model-theoretically. On the basis of the formalization, we demonstrate that a certain kind of soundness holds for this framework, while the completeness does not.

1 Introduction

The natural language inference (NLI) (Dagan et al., 2013) is the task of identifying the inferential relation between a text pair: a *premise* and *hypothesis*. If a hypothesis can be inferred from a premise using logical and commonsense knowledge, it is judged as *entailment*. If a premise and hypothesis are incompatible, it is judged as *contradiction*, and if neither of these cases hold, it is judged as *neutral*. NLI systems are expected to be applied to a wide range of fields, e.g., question answering, information retrieval, and text summarization.

In recent years, neural-based approaches (Parikh et al., 2016; Chen et al., 2017; Devlin et al., 2019; Lan et al., 2019; Tai et al., 2020; Wang et al., 2021) have achieved high accuracy in experiments with many NLI datasets, e.g., the SNLI corpus (Bowman et al., 2015), MultiNLI (MNLI) corpus (Williams et al., 2018), Adversarial NLI (ANLI) dataset (Nie et al., 2020), and QNLI dataset (Wang et al., 2018). However, neural-based models have the limitation that they cannot explain the reasoning processes by which inferential relations are derived. Such models represented a black box, and it is difficult to analyze what kind of inference was performed.[1] Yanaka et al. (2020) proposed a method to evaluate whether neural models learn the systematicity of monotonicity inference in natural language, and they demonstrated that the generalization ability of current neural models is limited. Gururangan et al. (2018) and Tsuchiya (2018) showed that NLI datasets such as the SNLI corpus and MNLI corpus have a hidden bias in that only a hypothesis is sufficient to determine inferential relations, and there is a risk that neural models are simply identifying inferential relations based on such biases.

From a different perspective, symbolic approaches to the NLI task have been proposed (Bar-Haim et al., 2007; MacCartney and Manning, 2007, 2008, 2009; Mineshima et al., 2015; Abzianidze, 2015, 2017; Hu et al., 2020). These approaches have the advantage that the reasoning process deriving the relation is understandable to humans, unlike neural-based approaches. In addition, symbolic approaches are generally founded on formal logic or linguistic analyses, which allows us to understand the reasoning processes. However, these approaches require inference rules that are created by humans; thus, it is difficult to handle synonyms, hypernyms, and hyponyms exhaustively. In addition, commonsense knowledge, e.g., “when it rains, the ground gets wet” must be expressed as inference rules; however, it is unlikely that such rules can be created exhaustively. Currently, these approaches have only been tested for the controlled NLI datasets such as the FraCaS test suite (Cooper et al., 1996) and SICK dataset (Marelli et al., 2014).

Saji et al. (2021) integrated these two approaches into a unified framework that can make the inference process explicit for certain linguistic phenomena while maintaining the applicability of neural-based approaches to relatively freely created (non-controlled) datasets. This method is based on the tableau method, which is one of the proof methods of formal logic; however, to the best of our

[1]One of the exceptions is the NLI system, which generates explanations by Kumar and Talukdar (2020); however, since they are generated by the neural model, the process of generating explanations is a black box.

knowledge, its theoretical limitations have not been clarified.

Thus, in this paper, we formalize the method proposed by Saji et al. model-theoretically to clarify the theoretical limitations of the method. We demonstrate that a certain kind of soundness holds for this method, while completeness does not.

2 Natural Language Inference using Neural Network and Tableau Method

This section provides an overview of the method proposed by Saji et al. For a premise $p \in L$ and hypothesis $h \in L$,[2] the method takes the following steps:

1. Construct the tableaux to prove entailment and contradiction relations.
2. Judge the closedness of the tableaux using a neural-based NLI system.
3. Determine the relation between the premise and hypothesis based on the constructed tableaux.

Below, we explain each step.

2.1 Tableaux Construction

The tableau method constructs a tree structure, referred to as a *tableau*, for a given set, each element of which is a pair of a sentence s and truth value v. Each node in a tableau is labeled with a tuple $e = (s, v, a, o)$ called an *entry*. Here, the entry e represents the constraint that s must take v as its truth value. a is the flag indicating whether the tableau rule (described below) has been applied to e, and o denotes the entry from which e was derived. The truth value v is either $\mathtt{T}$ or $\mathtt{F}$, representing true or false, respectively. The flag a is either 0 or 1, where 0 means that no tableau rule has been applied and 1 means that some rule has been applied. The *initial tableau* consists of entries for a given pair of natural language sentence and truth value, with $a = 0$. Each entry in the initial tableau is assumed to be derived from itself. That is, for each entry e in the initial tableau, e is in the form of $(s, v, 0, e)$. The tableau is created by repeatedly applying *tableau rules* to the entries. By applying a tableau rule to an entry, the constraint expressed by the entry is decomposed into several constraints. The decomposed constraints are added to the tableau as new entries. This decomposition makes the reasoning process explicit. In the following, when a tableau t' is derived by applying a tableau rule $r \in R$ to a tableau t, we write $t \overset{R}{\triangleright} t'$, and we refer to a tableau t which has no tableau t' such that $t \overset{R}{\triangleright} t'$ as a *complete tableau*.

Branches in a tableau indicate that there are multiple cases for possible valuation. A complete tableau to prove entailment relation whose initial tableau consists of $\{e_1 = (p, \mathtt{T}, 0, e_1), e_2 = (h, \mathtt{F}, 0, e_2)\}$ is referred to as an *entailment tableau* and that for contradiction whose initial tableau consists of $\{e_1 = (p, \mathtt{T}, 0, e_1), e_2 = (h, \mathtt{T}, 0, e_2)\}$ is referred to as a *contradiction tableau*.

A tableau rule takes the following form:[3]

$$r = (c_{1,1} \wedge \cdots \wedge c_{1,n_1}) \vee \cdots \vee (c_{m,1} \wedge \cdots \wedge c_{m,n_m}),$$

where $c_{1,1}, \ldots, c_{1,n_1}, \ldots, c_{m,1}, \ldots, c_{m,n_m}$ are functions that take $(s, v, 0, o)$ as input and return $(s', v', 0, o)$. Here, if $c_{i,j_i}(e)$ is defined for all c_{i,j_i} a tableau rule r is *applicable* to e. If a tableau has an entry e to which r is applicable, new m branches $\langle c_{1,1}(e), \cdots, c_{1,n_1}(e)\rangle, \cdots, \langle c_{m,1}(e), \cdots, c_{m,n_m}(e)\rangle$ are added as their children to all leaves dominated by the entry e. A tableau rule converts the constraint expressed in the source entry into the equivalent one.[4] There is no need to apply an operation to the entry e to which the tableau rule has been applied, because the constraint is already expressed by the entries derived from the entry e. The flag a in an entry (s, v, a, o) controls the application of such operations and is changed to 1 when a tableau rule is applied. An entry $(s, v, 1, o)$ is neither applied any tableau rules nor used in judging the closedness of the tableau as described below. Figure 1 (left) shows an entailment tableau for the following example:

Premise Either Smith or Anderson signed the contract.

Hypotheses If Smith did not sign the contract Anderson made an agreement.

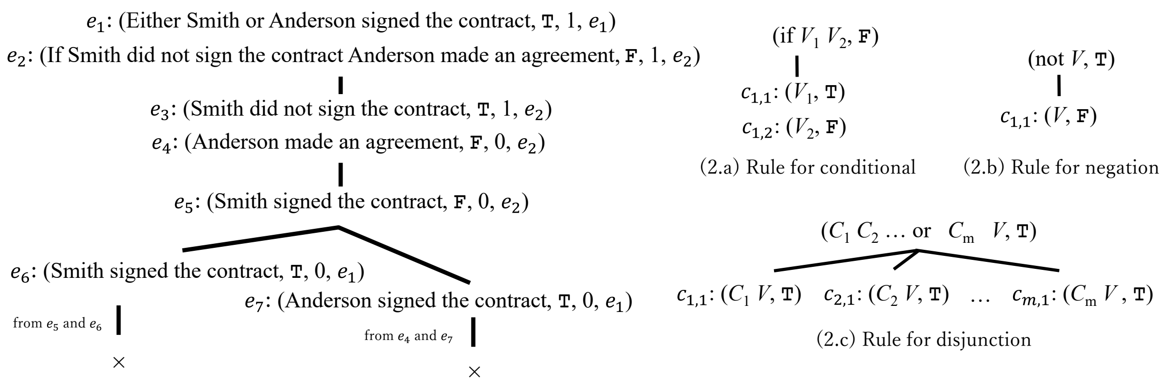
Label Entailment

The initial tableau of this example consists of e_1 = (Either Smith or Anderson signed the contract, $\mathtt{T}$, 0, e_1) and e_2 = (If Smith did not sign

[2] L is the set of natural language sentences.

[3] For a specific implementation of the tableau rules, see (Saji et al., 2021).

[4] Although tableau rules that derive weak constraints are also possible, such rules are not considered in the method proposed by Saji et al.



(1) Example of entailment tableau

(2) Examples of tableau rules

Figure 1: Example of entailment tableau in Saji et al.'s method

the contract Anderson made an agreement, F, 0, e_2). First, applying the rule (2.a) of Figure 1 to e_2 adds two new entries at the end of the path (the tableau leaf): $e_3 =$ (Smith did not sign the contract, T, 0, e_2) and $e_4 =$ (Anderson made an agreement, F, 0, e_2), and the flag of e_2 is changed to 1. Second, applying the rule (2.b) to e_3 adds a new entry: $e_5 =$ (Smith signed the contract, F, 0, e_2) and the flag of e_3 is changed. Finally, applying the rule (2.c) to e_1 adds two new entries: $e_6 =$ (Smith signed the contract, T, 0, e_1) and $e_7 =$ (Anderson signed the contract, T, 0, e_1), and the flag of e_1 is changed.

2.2 The Closedness of Tableaux

Saji et al. defined a branch b is *closed*, if and only if two entries $e_1 = (s_1, v_1, 0, o_1)$ and $e_2 = (s_2, v_2, 0, o_2)$ $(o_1 \neq o_2)$ on b satisfy one of the following three conditions, which we refer to as *closedness conditions*:

1. $v_1 = \mathtt{T} \wedge v_2 = \mathtt{F} \wedge s_1 = s_2$
2. $v_1 = \mathtt{T} \wedge v_2 = \mathtt{T} \wedge NLI(s_1, s_2) = \mathtt{C}$
3. $v_1 = \mathtt{T} \wedge v_2 = \mathtt{F} \wedge NLI(s_1, s_2) = \mathtt{E}$

Here, $NLI(s_1, s_2)$ is any NLI system that takes premise $s_1 \in L$ and hypotheses $s_2 \in L$ as inputs and returns one of the following classes: entailment (E), neutral (N), or contradiction (C). The first condition is similar to that of conventional tableau method, and the other two conditions are based on the NLI system. If all branches in a tableau are closed, the tableau is *closed*. The condition $o_1 \neq o_2$ excludes entries derived from only a premise or a hypothesis from the judging of the closedness; however, this is not a problem if a premise or hypothesis is neither tautology nor contradictory sentence.

Entailment tableau	Contradiction tableau	Output
Closed	Not closed	Entailment
Not closed	Closed	Contradiction
Not closed	Not closed	Neutral
Closed	Closed	Error

Table 1: Correspondence between closedness of tableaux and the inferential relation

As an example, let us consider the tableau shown in Figure 1 (left) and assume that $NLI\big(sen(e_7), sen(e_4)\big) = \mathtt{E}$.[5] The tableau has two branches. The left branch $\langle e_1, e_2, e_3, e_4, e_5, e_6 \rangle$ is closed because e_5 and e_6 satisfy the first closedness condition. The right branch $\langle e_1, e_2, e_3, e_4, e_5, e_7 \rangle$ is also closed because e_4 and e_7 satisfy the third closedness condition.

2.3 Determining the Inferential Relation

The inferential relation between a premise and a hypothesis is predicted based on the closedness of entailment and contradiction tableaux. This is identical to that of Abzianidze (2015) and summarized in Table 1.[6]

3 Model-Theoretic Formalization

3.1 Model

To discuss the method proposed by Saji et al. formally, we first define a model-theoretic interpretation of sentences. We then characterize tableau rules and NLI systems based on such interpretation, and define some properties of their proof system.

[5] Here, $sen\big((s, v, a, o)\big) = s$.

[6] The "error" class is for an uninterpretable situation where both entailment and contradiction relations hold.

A model is defined as follows:

Definition 1 (Model)**.** A model is a function from L to $\{\mathtt{T}, \mathtt{F}\}$. Let $\mathcal{M}$ be the set of all models. For a set of models $M \subseteq \mathcal{M}$, we define $M(s) = \{m \in M \mid m(s) = \mathtt{T}\}$.

Intuitively, a set of models $M(s)$ can be considered a set of situations where s is true. In the following, we define what NLI systems and tableau rules are consistent with a given set of models M,[7] and we clarify the theoretical limitations of the tableau methods of Saji et al. under the condition where the NLI system and tableau rules are consistent with the model set.

Definition 2. An NLI system is said to be *consistent* with a model set M if and only if all of the following conditions hold.

- $M(p) \subseteq M(h) \Leftrightarrow NLI(p, h) = \mathtt{E}$
- $\neg(M(p) \subseteq M(h)) \wedge \neg(M(p) \cap M(h) = \emptyset) \Leftrightarrow NLI(p, h) = \mathtt{N}$
- $M(p) \cap M(h) = \emptyset \Leftrightarrow NLI(p, h) = \mathtt{C}$

In this definition, the entailment relation corresponds to the inclusion relation of the model set, and the contradiction relation as the relation that the model sets are mutually disjoint.

In the following, we characterize the tableau methods based on the model-theoretic interpretation. In preparation, we define the model set for an entry, a branch of the tableau and a tableau by extending $M(s)$.

Definition 3 (Model set for an entry)**.** For a tableau entry $e = (s, v, a, o)$, we define $M(e)$ as follows:

$$M(e) = \begin{cases} M(s) & (v = \mathtt{T}) \\ M - M(s) & (v = \mathtt{F}) \end{cases}$$

Definition 4 (Model set for a branch)**.** For a branch b of a tableau, we define $M(b)$ as follows:

$$M(b) = \cap_{e \in b} M(e).$$

Definition 5 (Model set for a tableau)**.** Let B be a set of all branches in a tableau t. We define $M(t)$ as follows:

$$M(t) = \cup_{b \in B} M(b).$$

[7] The reason why we consider a subset M of $\mathcal{M}$ is that $\mathcal{M}$ contains logically unnatural models such as models that return $\mathtt{T}$ for every natural language sentence. We assume that the set of logically natural models is given as M. No conditions are required for M; thus, the following discussion is valid no matter what kind of M is.

Definition 6. Let $r = (c_{1,1} \wedge \cdots \wedge c_{1,n_1}) \vee \cdots \vee (c_{m,1} \wedge \cdots \wedge c_{1,n_m})$, E be the set of all entries, and $E_r = \{e \in E \mid r \text{ is applicable to } e\}$. A tableau rule r is said to be *consistent* with a model set M if and only if the following condition is satisfied for all $e \in E_r$:

$$M(e) = \Big(M\big(c_{1,1}(e)\big) \cap \cdots \cap M\big(c_{1,n_1}(e)\big)\Big) \cup \cdots \cup \Big(M\big(c_{m,1}(e)\big) \cap \cdots \cap M\big(c_{1,n_m}(e)\big)\Big).$$

In this equation, the left-hand side corresponds to the constraint of the entry e. The right-hand side represents the constraint derived by applying the tableau rule r to the entry e.

In addition, a tableau rule set R is said to be consistent with M if and only if all $r \in R$ are consistent with M.

3.2 Soundness and Completeness

The main components of the method proposed by Saji et al. are a tableau rule set R and NLI system NLI. We call a pair (R, NLI) *a proof system* and define the soundness and completeness of the proof system based on the model-theoretic interpretation. Furthermore, we prove any proof systems are sound and give a counterexample for the completeness. The soundness defined in this paper is the property that if the entailment (contradiction) tableau is closed, then the model sets for the premises and hypotheses are in an inclusion (mutually disjoint) relation. The completeness is the converse of the soundness. We define the soundness of a proof system as follows:

Definition 7 (Soundness)**.** Let M be a model set, R be a set of tableau rules consistent with M, and NLI be an NLI system consistent with M. We say that the proof system (R, NLI) is *sound* with respect to M if and only if the following condition holds for all premise p and hypothesis h:

- If the entailment tableau constructed by R is closed by NLI, then $M(p) \subseteq M(h)$.
- If the contradiction tableau constructed by R is closed by NLI, then $M(p) \cap M(h) = \emptyset$.

On the other hand, the completeness is defined as follows:

Definition 8 (Completeness)**.** Let M be a model set, R be a set of tableau rules consistent with M, and NLI be an NLI system consistent with M.

We say that the proof system (R, NLI) is *complete* with respect to M if and only if the following condition holds for all premise p and hypothesis h:

- If $M(p) \subseteq M(h)$, then the entailment tableau constructed by R is closed by NLI.
- If $M(p) \cap M(h) = \emptyset$, then the contradiction tableau constructed by R is closed by NLI.

3.2.1 Proof of Soundness

In this section, we prove the following theorem:

Theorem 1 (Soundness Theorem)**.** Let M be a model set. Any proof systems (R, NLI) consistent with M are sound with respect to M.

First, we introduce two lemmas to prove the theorem.

Lemma 2. Let M be a model set and R be a tableau rule set consistent with M. The following equation holds for any tableaux t and t' such that $t \overset{R}{\triangleright} t'$:

$$M(t) = M(t').$$

From Definition 6, Lemma 2 is trivial.

Lemma 3. Let t and t' be tableaux such that $t \overset{R^*}{\triangleright} t'$. If R is consistent with M, $M(t) = M(t')$.

Lemma 3 is trivial from Lemma 2.

The proof of Theorem 1 is as follows:

Proof. Assume that the entailment tableau t_{ent} constructed by R is closed by NLI. This implies that, two entries $(s_1, v_1, 0, o_1)$ and $(s_2, v_2, 0, o_2)$ $(o_1 \neq o_2)$ satisfy one of the closedness conditions in all branches b of t_{ent}.

- If the first closedness condition is satisfied, $v_1 = \mathtt{T} \wedge v_2 = \mathtt{F} \wedge s_1 = s_2$. Thus, from Definition 3, $M(e_1) \cap M(e_2) = M\big((s_1, \mathtt{T}, 0, o_1)\big) \cap M\big((s_1, \mathtt{F}, 0, o_2)\big) = M(s_1) \cap \big(M - M(s_1)\big) = \emptyset$.
- If the second closedness condition is satisfied, we obtain $v_1 = \mathtt{T} \wedge v_2 = \mathtt{T} \wedge NLI(s_1, s_2) = \mathtt{C}$. NLI is consistent with M; thus, from Definition 2, $M(s_1) \cap M(s_2) = \emptyset$. Therefore, from Definition 3, $M(s_1) \cap M(s_2) = M\big((s_1, \mathtt{T}, 0, o_1)\big) \cap M\big((s_2, \mathtt{T}, 0, o_2)\big) = M(e_1) \cap M(e_2) = \emptyset$.
- If the third closedness condition is satisfied, we obtain $v_1 = \mathtt{T} \wedge v_2 = \mathtt{F} \wedge NLI(s_1, s_2) = \mathtt{E}$. NLI is consistent with M; therefore, from Definition 2, $M(s_1) \subseteq M(s_2)$. Thus, from Definition 3, $M(e_1) = M(s_1)$, $M(e_2) = M - M(s_2)$. Since $M(s_1) \subseteq M(s_2)$, $M(e_1) \cap M(e_2) = \emptyset$.

From the above, $M(e_1) \cap M(e_2) = \emptyset$. Thus, from Definition 4, $M(b) = \emptyset$. Here, $M(b) = \emptyset$ for all branches $b \in t_{\text{ent}}$; thus, from Definition 5, $M(t_{\text{ent}}) = \emptyset$. Let t_{init} be the initial tableau. By $t_{\text{init}} \overset{R^*}{\triangleright} t_{\text{ent}}$ and Lemma 3, $M(t_{\text{init}}) = \emptyset$. Since t_{init} consists of $e_p = (p, \mathtt{T}, 0, e_p)$ and $e_h = (h, \mathtt{F}, 0, e_h)$, from Definition 4 and 5, $M\big((p, \mathtt{T}, 0, e_p)\big) \cap M\big((h, \mathtt{F}, 0, e_h)\big) = \emptyset$. Thus, $M\big((p, \mathtt{T}, 0, e_p)\big) \subseteq M\big((h, \mathtt{T}, 0, e_h)\big)$. From Definition 3, $M(p) \subseteq M(h)$. The above satisfies the first condition of Definition 7. The second condition (for the contradiction tableau) can be proved in a similar way. □

3.2.2 Counterexample of Completeness

In this section, we present a counterexample such that a proof system is not complete. Let us consider the following example:

Premise Smith and Anderson did not go out.

Hypothesis Smith and Anderson were home.

Label Entailment

The entailment tableau shown in Figure 2 is for the premise and hypothesis. The truth values of entries e_4, e_5, e_6 and e_7 to which the tableau rule has not yet been applied, are all $\mathtt{F}$. In the closedness conditions, the truth value of one of the entries must be $\mathtt{T}$; thus, this entailment tableau cannot be closed with any NLI. As an example of the model set M, let us consider the models shown in Table 2. In this model set M, $M(e_1) = M(e_3) = M(e_4) \cap M(e_5) = \{m_1\}$ and $M(e_2) = M(e_6) \cup M(e_7) = \{m_2, m_3, m_4\}$; thus, these tableau rules are consistent with M. Note that $M(p) \subseteq M(h)$ since $M(p) = M(h) = \{m_1\}$. This means that the first condition of Definition 8 does not hold. Thus, the completeness does not hold.

3.3 Example of Complete Proof System

Generally, the completeness does not hold; however, we can prove that a certain proof system is complete. In this section, we present an example of such proof system.

Let M be a model set, NLI be an NLI system that is consistent with M and R be a tableau rule

e_1: (Smith and Anderson did not go out, T, 1, e_1)
e_2: (Smith and Anderson were home, F, 1, e_2)

e_3: (Smith or Anderson went out, F, 1, e_1)

e_4: (Smith went out, F, 0, e_1)
e_5: (Anderson went out, F, 0, e_1)

e_6: (Smith was home, F, 0, e_2) e_7: (Anderson was home, F, 0, e_2)

Figure 2: Counterexample of completeness

		m_1	m_2	m_3	m_4	M
Smith and Anderson did not go out	$\neg(\neg S \vee \neg A)$	T	F	F	F	$\{m_1\}$
Smith and Anderson were home	$S \wedge A$	T	F	F	F	$\{m_1\}$
Smith or Anderson went out	$\neg S \vee \neg A$	F	T	T	T	$\{m_2, m_3, m_4\}$
Smith went out	$\neg S$	F	F	T	T	$\{m_3, m_4\}$
Anderson went out	$\neg A$	F	T	F	T	$\{m_2, m_4\}$
Smith was home	S	T	T	F	F	$\{m_1, m_2\}$
Anderson was home	A	T	F	T	F	$\{m_1, m_3\}$

Table 2: Models for counterexample of completeness

set consistent with M such that for all $r \in R$, the following conditions hold:[8]

1. r is in the form of $c_{1,1}$.
2. If $c_{1,1}\big((s, v, 0, o)\big)$ is defined, the returned value is in the form of $(s', v, 0, o)$.

In this proof system, entailment tableaux constructed by R are always closed by NLI if $M(p) \subseteq M(h)$. The proof of its completeness is as follows:

Proof. Assume that $M(p) \subseteq M(h)$. Here, let t_{init} be the initial tableau constructed from $e_p = (p, \mathtt{T}, 0, e_p)$ and $e_h = (h, \mathtt{F}, 0, e_h)$. For all tableaux t such that $t_{\text{init}} \overset{R^*}{\rhd} t$, the following statements hold:

- t has only one branch.
- t has only two entries whose flag is 0: the one is in the form of $(p', \mathtt{T}, 0, e_p)$ and the other is in the form of $(h', \mathtt{F}, 0, e_h)$.
- $M(p) = M(p')$
- $M(h) = M(h')$

The first statement holds from the first condition about R. The second statement holds from the conditions about R. From the conditions about R, Definition 3 and 6, the last two statements hold. The entailment tableau t_{ent} derived from t_{init} also satisfies the above statements, since $t_{\text{init}} \overset{R^*}{\rhd} t_{\text{ent}}$. Because $M(p') = M(p) \subseteq M(h) = M(h')$ and NLI is consistent with M, $NLI(p', h') = \mathtt{E}$. Thus, the branch of t_{ent} satisfies the second closedness condition, that is, t_{ent} is closed by NLI. The second condition (for the contradiction tableau) of the completeness also can be proved in the similar way. Thus, this proof system is complete with respect to M. □

[8]We can consider that the tableau rules simply paraphrase sentences.

4 Conclusion

In this paper, we have formalized the method proposed by Saji et al. based on a model-theoretic interpretation and have clarified the theoretical limitations of this method. We have proved the soundness theorem and provided an example of complete proof system. In future work, we will explore what kind of proof systems are complete.

Acknowledgments

This research was partially supported by the Grant-in-Aid for Scientific Research (C) (No. 22K12148) of JSPS.

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.

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. 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.

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. 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. Association for Computational Linguistics.

Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Josef Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, Steve Pulman, Ted Briscoe, Holger Maier, and Karsten Konrad. 1996. Using the framework.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. *Recognizing Textual Entailment: Models and Applications*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

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.

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. 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.

Sawan Kumar and Partha Talukdar. 2020. NILE : Natural language inference with faithful natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8730–8742, Online. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

Bill MacCartney and Christopher D. Manning. 2007. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200, Prague. Association for Computational Linguistics.

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. Coling 2008 Organizing Committee.

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

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

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.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational*

Linguistics, pages 4885–4901, Online. Association for Computational Linguistics.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas. Association for Computational Linguistics.

Ayahito Saji, Daiki Takao, Yoshihide Kato, and Shigeki Matsubara. 2021. Natural language inference using neural network and tableau method. In *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*, pages 406–414, Shanghai, China. Association for Computational Linguistics.

Wen Tai, H. T. Kung, Xin Dong, Marcus Comiter, and Chang-Fu Kuo. 2020. exBERT: Extending pre-trained models with domain-specific vocabulary under constrained training resources. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1433–1439, Online. Association for Computational Linguistics.

Masatoshi Tsuchiya. 2018. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma. 2021. Entailment as few-shot learner. *ArXiv*, abs/2104.14690.

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.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, and Kentaro Inui. 2020. Do neural models learn systematicity of monotonicity inference in natural language? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6105–6117, Online. Association for Computational Linguistics.