修辞構造に基づく分割統治型 LLM 翻訳

田中 邦朋 [1]　帖佐 克己 [2]　平尾 努 [2]　笹野 遼平 [1]
[1] 名古屋大学大学院 情報学研究科　[2] NTT コミュニケーション科学基礎研究所
tanaka.kunitomo.z3@s.mail.nagoya-u.ac.jp　katsuki.chousa@ntt.com
tsutomu.hirao@gmail.com　sasano@i.nagoya-u.ac.jp

概要

大規模言語モデルの急速な発展は、機械翻訳分野にも影響を及ぼしているが、長く複雑な文の翻訳においては、原文構造の喪失、訳抜けといった課題が依然として存在する。本研究では、大規模言語モデルの特長を活かしつつ、これらの課題に対処するため、原文の修辞構造を利用した分割統治型機械翻訳法を提案する。提案手法では、原語文を修辞構造に則って分割した後、モデルによる翻訳と原文の並列構造を考慮した統治を行って、翻訳文を得る。特許請求項の日英翻訳タスクを通して評価を行った結果、翻訳精度、訳抜け抑制、文構造保持の点で、提案手法が大規模言語モデルでの直接翻訳より優れていることが確認された。

1　はじめに

近年、大規模言語モデル（Large Language Model; LLM）の発展により、自然言語処理の様々なタスクにおいて著しい性能向上がみられている [1]。その中でも機械翻訳は、LLM の主要な応用分野の一つとして注目を集めており [2]、多くの場面で実用的な性能に達しているが、課題も存在する [3]。特に、長く構造が複雑な文における翻訳では、原文の構造が適切に保持されない、訳抜けが生じるといった問題が表れやすい [4]。こうした課題は、翻訳対象のドメインによっては致命的な問題となる。

本研究では、これらの課題に対処するため、修辞構造を活用した、新しい分割統治アプローチによる翻訳手法を提案する。提案手法の概略を図 1 に示す。まず、修辞構造木に基づいて用言並列を展開し、長く複雑な一文を短文に分解する。次に、各短文を LLM に入力して個別に翻訳し、翻訳短文を得る。最後に、並列の入れ子の深さに従って、分割翻訳文の統合の順番を決定し、LLM を用いた統合を繰り返すことで、最終的な翻訳を得る。このとき、翻訳短文に対応する原語文を活用することで、翻訳精度の向上を図る。この手法には主に二つの利点がある。第一に、原文を修辞構造に基づき分割することで個々の翻訳対象を短くしつつも、意味を維持したより正確な翻訳が可能となる。特に、用言並列の展開により、統治するべき短文の数を抑えながら、長大な並列句を分解できる。第二に、分割された短文を並列構造を保持するように統合することで、原文の構造と整合した翻訳を実現できる。さらに、提案手法は LLM の追加学習を伴わず、特定のモデルに制約されない。このような特徴により、導入が簡便で、より強力な LLM への応用が可能である上、モデル入力の設計に着目することが、LLM を用いた翻訳において有効な手法となりうることを示す。

提案手法の有効性を示すため、本研究では、特許請求項の日英翻訳タスクに着眼して、LLM による直接翻訳との比較評価を行った。特許請求項は、一文が長い上に、階層的な並列構造を多く含む複雑な構造を持つという特徴があるため、LLM を用いた翻訳でもあっても、誤訳が現れやすいことが指摘されている [5]。評価実験では、LLM による原文の直接翻訳と提案手法による翻訳を対象に、自動評価指標を用いた比較分析を実施した。その結果、長大で複雑な構造を持つ文の翻訳において、提案手法は、翻訳精度、訳抜けの抑制、および文構造の保持の観点から、直接翻訳よりも優れていることが確認された。

2　関連研究

LLM の急速な発展は、機械翻訳分野にも大きな影響を与えており、従来手法では扱うことが難しかった高度な翻訳を可能にしつつある [2]。しかしながら、長く複雑な構造を持つ文の翻訳は、LLM を用いても依然として困難である [4]。特に特許請求項は、現状の LLM による翻訳では実用性に課題が残ることが指摘されている [5]。そこで本研究では、LLM への入力となる原語文に対し、分割統治的な機械翻

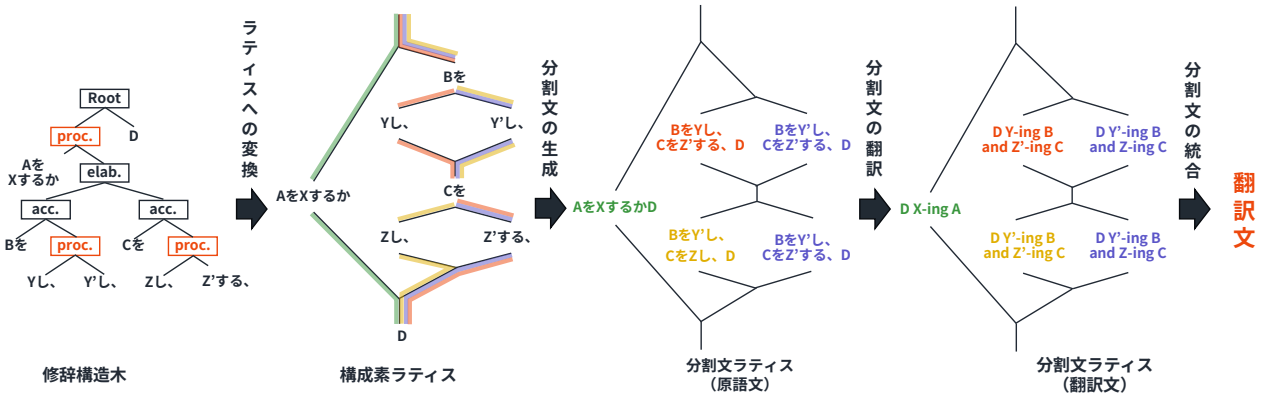

図 1　提案手法の概略

訳のアプローチを適用する手法を提案し、特許請求項の日英翻訳タスクで手法を評価する。

分割統治的な機械翻訳は、原語文をより細かい単位に分割し、それぞれを翻訳した上で、これらを統合して最終的な翻訳文を生成する翻訳手法である。長文を翻訳する際の訳抜けや繰り返しの低減が期待され、分割単位や統合方法を工夫した種々の研究が行われてきた。具体的には、線形計画問題として原語文を分割したのち、個別に翻訳してそのまま結合する手法 [6] や、節や等位接続節を分割単位に、翻訳された各分割の間に特殊トークンを差し込んで結合してから、並べ替えと編集を学習済モデルで行うもの [7, 8] が提案されている。これらの既存手法は、原言語文を連続した部分文字列の集合へと分割する点に特徴がある。そのため、翻訳後の統治ステップにおいて、分割された翻訳文の並べ替えとそれに伴う修正が必要となる。

本研究では、原文の修辞構造木を活用して、原文から含意される短文を生成する。その後の統治ステップでは、並列構造に従って翻訳された短文を順次統合して翻訳文を得ることで、並べ替えを必要としない、より簡潔な翻訳を実現する。

3　手法

本研究では、分割統治翻訳の枠組みに則り、原語文を修辞構造に基づいて分割した後、LLM を用いて、翻訳と原文の並列構造を考慮した統治を行って、翻訳文を得る[1)]。本手法の概略を図 1 に示す。

3.1　並列構造を反映したラティスの構築

非終端ノードに修辞構造ラベル、終端ノードに構成素をもつ修辞構造木に基づき、原文の用言並列構造を反映したラティスを構築する[2)]。ここで本処理では、修辞構造木のノードを深さ優先探索で文頭側から走査する。終端ノードに到達するたびにラティスにノードを追加し、木の終端ノードが保持する構成素を格納する。

また、用言並列ラベルを持つ非終端ノードを検出した場合、そのノードを起点とする並列構造を同定し、各並列構造が共通のノードから分岐を開始して共通の終点ノードで分岐を終了するようにラティスを構築する。この時の分岐を終了するノードを合流ノードと定義し、その並列構造の入れ子の深さを記録する。以下では、ここで完成したラティスを構成素ラティスと呼ぶ。

3.2　分割

3.1 節で構築した構成素ラティスでは、ソースノードからシンクノードに至る各パスが、原語文から含意される分割原語文に対応する。しかし、単純に全てのパスを考慮すると、部分パスの重複することに加え、並列構造の連続によってパス数が爆発的に増大する。

そこで、少数でありながらも全てのノードを考慮できるパス集合を得ることを考える。まず、構成素ラティスから、他の並列構造を包含しない最も深い階層にあるノードを特定する。以下、ここで得られるノード集合の元を最深ノードと呼ぶ。ソースノードから深さ優先探索でラティスのノードを走査し、3.1 節でマークした合流ノードが現れた時、その先行ノードを最深ノードとして認定する。さらに、縦方向に連なる並列を特定するため、認定した最深ノードよりも入れ子が浅い合流ノードが出てくるまでに合流ノードが見つかった場合にも、最深ノード

1)　LLM に入力した指示については A 節を参照。

2)　構成素ラティスの構築アルゴリズムの詳細を B 節に付す。

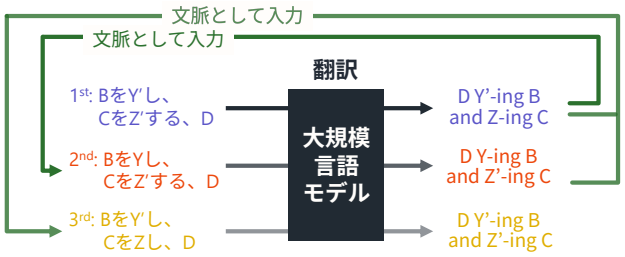

図 2　LLM を用いた逐次的な翻訳処理

と認定する。

次に、各最深ノードに着目し、分割文に対応するパスを取得する。ソースノードから各最深ノードへ、各最深ノードからシンクノードへと至るパスを、なるべく文末に近い構成素を持つノードを経由するように選択する。日本語では、必ず文頭から文末側へ語句が係るため、文末に近い並列を採ることで、生成される短文内における係り受け表現がより自然に保たれる。そして、得られたパスとそれに対応する分割原語文を、各最深ノードへ格納する。

このような分割手法により、どの用言並列に着目した分割文であるかを保持しながら、なるべく自然な分割原語文を獲得する。さらに、分割原語文に並列構造が付随していることで、後段のステップで分割翻訳文を統治する順番を合理的に決定できる。

ここで構築したラティスを、3.1 節の構成素ラティスとは区別して、分割文ラティスと呼ぶ。

3.3　翻訳

翻訳ステップでは、分割文ラティスに含まれる各文を LLM に翻訳の指示とともに入力し、分割文の系列を翻訳文へ変換する。まず、分割文ラティスに含まれる各分割原語文を、文末に現れる並列に着目してできたものから順に取得する。これらを翻訳の指示とともに LLM に順に入力し、翻訳された分割翻訳文へと翻訳したものを対応する分割原語文があったノードへと格納する。この際、分割文の翻訳対は、次のノードの翻訳処理以降、翻訳例として文脈へ追加される。この様子を図 2 に示す。なお、文末に近い順で分割原語文のノードを取り出すのは、分割原語文の生成時と同様の理由で、先により自然な生成文を取り出せるからである。これにより、先に処理された分割原語文とその翻訳が文脈にも例示として与えられるため、自然な文がなるべく多くの翻訳で利用されることを期待する。

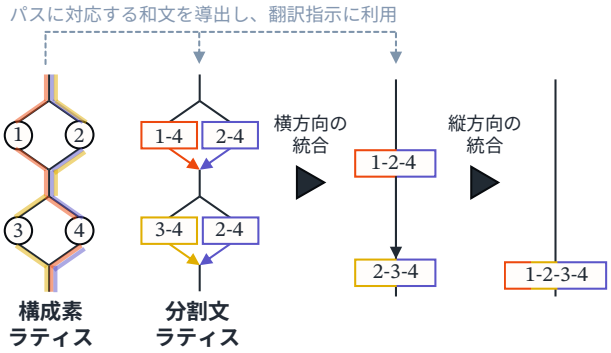

図 3　各反復での統合処理。数字は、構成素ラティスでのノード番号を指し、原語文の各構成素に対応する。

3.4　統治

最後に、分割文ラティスに含まれる分割翻訳文を順に統合することで、最終的な翻訳文を獲得する。統治ステップでは、分割文ラティスのシンクノードから、深さ優先後行順でラティスを走査することで入れ子の深さごとに合流ノードの順序付きリストを得て、入れ子が深い順に分割翻訳文を反復して統合する。

各反復においては、まず横方向の並列の統合を行い、続いて縦方向の並列の統合を行う。各反復での統合処理の概略を図 3 に示す。横方向の並列の統合では、3.2 節での分割文ラティスの構造に従い、各合流ノードの先行ノードにある分割翻訳文を統合する。生成された分割翻訳文は各合流ノードに格納する。その後、同じ入れ子の深さで縦方向に連なる並列を検出し、これらを統合して文末側の合流ノードに格納する。

統合の各ステップでは、分割翻訳文とそれに対応する分割原語文の対を例示として LLM に与えながら、合流ノードに対応する分割原語文を入力する。翻訳結果という形で得られた分割翻訳文は、次以降の統合で対応する分割原語文とともに文脈として与えられ、かつ合流ノードに格納される。ここで、各部分翻訳に対応する分割原語文は、3.2 節で各合流ノードに持たせたパス情報をもとに、3.1 節で構築した構成素ラティスから生成する。さらに、統合の際に、分割文に対応するパスに含まれるノード集合の和集合を取って合流ノードへ格納することで、分割翻訳文と分割原語文が常に対応づくようにする。

このようにして生成された分割翻訳文は、翻訳ステップと同様、対応する分割原語文とともに、次ステップ以降で翻訳の例示として与えられる。

4 実験

本研究では、以下に示す実験を行い、提案手法の有効性を確認した。

4.1 実験設定

本研究では、評価データとして JaParaPat [9] に含まれる特許請求項の日英翻訳対のうち、1788 例を用いた。原語文の修辞構造解析器には、Kobayashi らの手法 [10] に対し、新森らの規約 [11] に従い特許請求項に修辞構造のアノテーションを与えたデータセットを用いて学習した解析器を用いた。翻訳および統治の際に利用する LLM として gpt-4o-20240806 [1] を選定した。なお、LLM については、出力に決定性を持たせるため、温度定数 `temperature` と、核抽出法の閾値 `top_p` を 0 に設定した。また、ベースラインとして、上記設定の LLM に翻訳を指示しながら原語文全体を入力して翻訳する手法を採用した[3)]。

4.2 評価指標

得られた翻訳について、翻訳の質の自動評価と、原語文との構造の整合性に関する評価を行った。翻訳の質の自動評価には COMET [12] を用いた。構造の自動評価は、StrAlign [13]、および LLM による含意関係認識によって行った。StrAlign は、2 つの文間の類似度を双方の句構造木から得られる部分木 (スパン) の一致に基づき計算する。なお、部分木の一致はスパンの埋め込みベクトルの類似性を手がかりとする[4)]。この手法により、提案手法で得られた翻訳英文と、それに対応する評価データの参照英文との統辞的な類似度を評価する。また、提案手法による翻訳が、分割原語文の内容を含んでいるかを評価するために、LLM による言語横断的な含意関係認識 (Recongizing Textual Entailment; RTE) を行った。具体的には、LLM に含意関係認識をするよう指示を与え [3)]、提案手法による翻訳文が、それに対応する原語文から生成された分割原語文の内容を含意するかを判定させる[5)]。LLM が含意と判定すれば、分割原語文の内容が抜けずに翻訳文に含まれていることとなる。本稿では、分割原語文全体のうち、対応する翻訳文に含意されると判定された割合を報告する。

表 1 各手法による翻訳結果の評価結果

評価指標	COMET	StrAlign (%)			RTE (%)
		F1	Prec.	Rec.	
提案手法	**0.8065**	**51.7**	**47.8**	56.2	**93.71**
ベースライン	0.8021	50.7	45.7	**57.1**	91.09

4.3 実験結果

各指標による評価結果を表 1 に示す。

COMET について、提案手法はベースライン手法と比べてスコアが 0.4 パーセントポイント高く、提案手法により翻訳の質が改善されていることがわかる。StrAlign では、F1 値がベースラインより 1.0 ポイント高いことに加え、適合率がベースライン手法を 2.1 ポイント上回っている。この結果は、提案手法による翻訳に含まれる統辞的な構造が、参照英文の構造をより忠実に反映していることを示している。さらに、LLM による含意関係認識では、提案手法による翻訳の方がベースライン手法よりも 2.6 ポイントほど多く含意していると判定された。提案手法による翻訳は、原語文に含まれる内容を失わずに反映しており、ベースライン手法と比較して訳抜けが少ないといえる。

5 おわりに

本研究では、修辞構造に基づいた文の分解と、LLM を用いた再帰的な翻訳・統合により、複雑な構造を持つ文を翻訳する方略を提案した。特許請求項の日英翻訳を通した評価結果から、品質、構造維持、訳抜けの少なさにおいて、提案手法が直接翻訳より優れていることが示された。しかし、本研究の提案手法では、分割文の生成時に構成素の末尾が次の要素へ正しく係り受けしない場合を考慮しておらず、これを改善することで翻訳のさらなる改善が見込まれる。また、文末に近い構成素に着目する手法は、規範的な日本語文では修飾が一貫して前置修飾の形式をとるという、言語に依存した特徴を利用しており、多言語への拡張が考慮されていない。この 2 点は今後の研究課題としたい。

3) LLM に入力した指示については A 節を参照。
4) Evalb (https://nlp.cs.nyu.edu/evalb/) におけるスパンの一致をソフトマッチングにしたものと考えれば良い。
5) LLM による RTE のメタ評価については、C 節を参照。

参考文献

[1] OpenAI. GPT-4o System Card. **arXiv preprint arXiv:2410.21276**, 2024.

[2] Chenyang Lyu, Zefeng Du, Jitao Xu, Yitao Duan, Minghao Wu, Teresa Lynn, Alham Fikri Aji, Derek F. Wong, and Longyue Wang. A Paradigm Shift: The Future of Machine Translation Lies with Large Language Models. In **Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)**, pp. 1339–1352, 2024.

[3] Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, et al. Findings of the WMT24 General Machine Translation Shared Task: The LLM Era Is Here but MT Is Not Solved Yet. In **Proceedings of the Ninth Conference on Machine Translation (WMT)**, pp. 1–46, 2024.

[4] Ananya Mukherjee, Saumitra Yadav, and Manish Shrivastava. CoST of breaking the LLMs. In **Proceedings of the Ninth Conference on Machine Translation (WMT)**, pp. 299–306, 2024.

[5] Lekang Jiang, Caiqi Zhang, Pascal A Scherz, and Stephan Goetz. Can Large Language Models Generate High-quality Patent Claims? **arXiv preprint arXiv:2406.19465**, 2024.

[6] Jean Pouget-Abadie, Dzmitry Bahdanau, Bart van Merriënboer, Kyunghyun Cho, and Yoshua Bengio. Overcoming the curse of sentence length for neural machine translation using automatic segmentation. In **Proceedings of Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST)**, pp. 78–85, 2014.

[7] 加納保昌, 須藤克仁, 中村哲. 分割統治的ニューラル機械翻訳. 言語処理学会 第 27 回年次大会 発表論文集, 2021.

[8] 石川隆太, 加納保昌, 須藤克仁, 中村哲. 文内コンテキストを利用した分割統治ニューラル機械翻訳. 言語処理学会 第 29 回年次大会 発表論文集, 2023.

[9] Masaaki Nagata, Makoto Morishita, Katsuki Chousa, and Norihito Yasuda. JaParaPat: A large-scale Japanese-English parallel patent application corpus. In **Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)**, pp. 9452–9462, 2024.

[10] Naoki Kobayashi, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. A simple and strong baseline for end-to-end neural RST-style discourse parsing. In **Findings of the Association for Computational Linguistics (EMNLP)**, pp. 6725–6737, 2022.

[11] 新森昭宏, 奥村学, 丸川雄三, 岩山真. 手がかり句を用いた特許請求項の構造解析. 情報処理学会論文誌, Vol. 45, No. 3, pp. 891–905, 03 2004.

[12] Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. COMET: A neural framework for MT evaluation. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 2685–2702, 2020.

[13] Katsuki Chousa and Tsutomu Hirao. Automatic Evaluation of Language Generation Technology Based on Structure Alignment. In **Proceedings of the 31st International Conference on Computational Linguistics (COLING)**, 2025.

A　各手順で用いたプロンプト

この節では、各手法で用いたプロンプトを示す。赤字がシステムプロンプト、青字が指示部、緑字が例示部における応答部を指す。なお、出力が逐次的に文脈に与えられる場合は、指示部と応答部が文脈に付け加えられてから、次の応答を出力させるためにもう一度指示部が与えられる。

A.1　翻訳

```
Follow the instruction faithfully, and output only what it requires.  No other response is allowed.  Also, you are required to be consistent with the translation of each terminology throughout trials.
Translate a following patent claim in Japanese into English: { 分割原語文 }
{ 分割翻訳文 }
```

A.2　統治

```
Follow the instruction faithfully, and output only what it requires.  NO OTHER RESPONSE IS ALLOWED.
Translate a given Japanese noun phrase into English in one sentence.
Japanese: { 分割原語文 }
English:  { 分割翻訳文 }
```

A.3　含意関係認識

```
Follow the instruction faithfully, and output only what it requires.  NO OTHER RESPONSE IS ALLOWED.
Given a premise in English, judge whether it entails a Japanese hypothesis.  If so, return 'Entailment'. If not, return 'Not entailment'.
Premise: { 翻訳文 }
Hypopthesis: { 分割原語文 }
{Entailment / Not Entailment}
```

A.4　直接翻訳ベースライン

```
Follow the instruction faithfully, and output only what it requires. No breakline is allowed.
Translate a following patent claim in Japanese into English. You are required to keep the result to be a noun phrase:{ 原語文 }
{ 翻訳文 }
```

B　構成素ラティス構築アルゴリズム

Input: 修辞構造木 (2 分木) $T = (V_T, E_T)$
Output: 平面ラティス $L = (V_L, E_L)$

Function `ConstructSublattice`(T)**:**
- $v_l \leftarrow \text{newnode}(V_L)$;　$V_L \leftarrow v_l \cup V_L$;
- **for** $v_t \in V_T$ **in preorder do**
  - $v'_l \leftarrow \text{newnode}(V_L)$;　$V_L \leftarrow \{v'_l\} \cup V_L$;
  - **if children**$(v_t) = \emptyset$ **then**
    - v_l.text $\leftarrow v_t$.text;
    - $E_L \leftarrow \{(v_l, v'_l)\} \cup E_L$;
    - $v_l \leftarrow v'_l$;
  - **else if** v_t**.label = "procedure" then**
    - $P \leftarrow \text{subtree}(v_t.\text{left}), \text{subtree}(v_t.\text{right})$
    - $p' \leftarrow \text{subtree}(v_t.\text{right.right})$;
    - **while** p'**.label = "procedure" do**
      - $P \leftarrow P \cup \text{subtree}(p'.\text{left})$;
      - $p' \leftarrow \text{subtree}(p'.\text{right})$;
    - **for** $T' \in P$ **do**
      - $L' =$ `ConstructSublattice`(T');
      - $E_L \leftarrow \{(v_l, L'.\text{source}), (L'.\text{sink}, v'_l)\} \cup E_L$;
    - $v_l \leftarrow v'_l$;
    - $V_L \leftarrow V_L \cup \{v_l\}$;
- **return** L;

図 4　修辞構造木から構成素ラティスへの変換

C　LLM による RTE のメタ評価

本研究では、原語文の各要素が翻訳文に含まれるかを見るために、翻訳文が対応する原語文から得られる各分割原語文を含意するかを LLM で評価した。その正当性を確認するため、分割原語文と翻訳文のペアを 100 例を無作為抽出し、著者らによる分類を行った。表 2 に、無作為抽出された 100 例に対する人手での分類結果と LLM による分類予測を示す。

表 2　RTE の人手分類と LLM による予測の結果

true\pred	Not Entailment	Entailment	合計
Not Entailment	6	10	16
Entailment	1	83	84
合計	7	93	100