

日本語推論システム lightblue の開発環境構築に向けて

佐伯小遥 富田朝 戸次大介
お茶の水女子大学

{g2120517, tomita.asa, bekki}@is.ocha.ac.jp

概要

本論文では、Haskell の Web アプリケーションフレームワーク Yesod を用いて、日本語推論システム lightblue による文の解析結果や推論結果を可視化する文法開発環境 express の改良を行った。特に、型検査証明図や推論証明図を可視化し、解析結果や推論結果の構造を直感的に理解可能とした。また、ユーザが証明図を操作できる機能を実装し、解析プロセスを詳細に検討できるようにした。加えて、証明が失敗した場合には、その情報をユーザに明示することで、エラーの特定とデバッグ作業を効率化した。本論文では、express のシステム概要と実装についての詳細や、実使用に向けた今後の課題を示す。

1 はじめに

自然言語推論 (Natural Language Inference; NLI) は自然言語処理における重要なタスクである。NLI には大規模言語モデル (LLM) によるアプローチと、理論言語学に基づく計算言語学的アプローチが存在する。前者のアプローチでは、推論の対象になる範囲が急速に拡大しているが、LLM がその推論過程を説明することは難しく、また生成 AI による説明が正しいという保証はない。一方で、後者の計算言語学的なアプローチに基づく自動推論では、推論結果を出力するだけでなく、その推論の妥当性を保証する証明図を出力することができるため、説明性において利点がある。

計算言語学に基づく自動推論システムとして、ccg2lambda [1] や lightblue [2] があげられる。lightblue は、意味の理論に依存型意味論 (Dependent Type Semantics; DTS) [3] を採用している。lightblue に依存型理論 (Dependent Type Theory; DTT) [4] の定理自動証明器 wani[5, 6] を接続することで、照応や前提束縛などを含む複雑な言語現象の解析を実現している。

近年、lightblue を用いて構築された日本語 CCG ツリーバンクの言語学的妥当性の評価 [7, 8] や、日本語テストセットにおける lightblue の推論結果の評価およびエラー分析 [9] など、lightblue の性能を向上させる研究が進められている。その中において、現在次のような課題が生じている。

- 解析結果や推論結果の可視化が行われていないため、結果の詳細な把握が困難であること
- エラーの発生箇所や原因の特定を行うための機能がないため、開発効率が低いこと

これらの課題により、既存の可視化環境では、解析結果を実装のデバッグや理論の見直しに十分活用することができない。そこで、本論文では日本語推論システム lightblue の開発環境である express [10] の改良に取り組む。特に、解析結果の視認性を上げ、開発効率を向上させるようなユーザインタフェースを提案し、開発環境の課題解決を目指す。

2 背景

2.1 自然言語推論

NLI は、与えられた前提と仮説間に含意関係が成り立つかを判定するタスクである。前提と仮説の間に含意関係が成立する場合には含意 (YES)、前提と仮説の否定の間に含意関係が成立する場合には矛盾 (NO)、そのいずれでもない場合には中立 (UNK) と判定する。例を以下に示す。

前提： 山田先生が青木さんを叱った。

仮説： 青木さんが山田先生に叱られた。

この例では、前提文「山田先生が青木さんを叱った」が真であるようなすべての状況で、仮説文「青木さんが山田先生に叱られた」が真であるので、前提が仮説を含意する。

2.2 日本語推論システム lightblue

lightblue は関数型プログラミング言語 Haskell で実装された統語論と意味論の分析を統合した日本語推論システムである。

統語理論に組合せ範疇文法 (Combinatory Categorical Grammar; CCG) [11, 12] を、意味理論に依存型意味論を採用しており、日本語の文を入力として受け取ると CCG 統語構造や DTS 意味表示を出力する。

加えて、入力文に対して、意味表示の型は依存型理論における type であることを調べる型検査、そして定理自動証明器 wani を用いて先述した自然言語推論を行い、それらが成功している場合にはその証明図を出力するという特徴を持つ [13]。この出力される証明図によって、lightblue による推論結果、型検査結果の妥当性が保証される。

2.3 課題：文法開発における可視化

このように lightblue の実装は進んでいるが、文法開発は複雑な過程である。高度な専門性を要するばかりか、非効率な側面もあるため、文法開発過程の可視化が課題となっている。一般的に文法開発の流れは図 1 のようなサイクルを辿る。lightblue の解析結果を分析・評価することによって、CCG や DTS の理論が修正されたり、lightblue や定理自動証明器 wani の実装が見直されることが理想である。

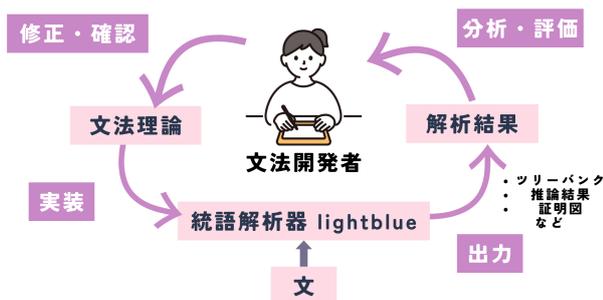


図 1 文法開発の流れ

しかしながら、現在 lightblue には、文法開発のために解析結果を分析しやすいような可視化環境が用意されていないため、実装のデバックや理論の見直しに解析結果を活かすことが難しい。例えば現在以下のような課題がある。

- 統語構造や型検査証明図、推論証明図を html 形式で一度に全て出力しているため、分析対象の解析結果にアクセスしづらい

- 統語構造や証明図の全体図が画面に入り切っておらず、構造を把握しづらい
- デバックを支援する機能がないため、開発効率が低い

lightblue の HTML 表示では、解析結果が画面の右や下に広がる形式で提示されており、ユーザーはスクロール操作を通じて全体を確認する必要がある。このような出力形式では、解析結果の全体像を直感的に把握しにくく、結果の詳細な分析においてユーザーの負担が大きくなるという問題がある。既存の lightblue の解析結果の出力の様子は付録図 7 に示す。

3 先行研究

GUI を持つ文法開発プロジェクトとして、express, GMB Explorer, PMB Explorer を取り上げる。

3.1 express

GUI を備えた既存の lightblue 開発システムとして express がある。express では、JSeM (Japanese semantic test suite: 日本語意味論データセット) [14] に対する統語構造・意味表示をブラウザ上に表示することができる。JSeM とは、日本語の意味的な現象に基づく含意関係のデータセットである。

lightblue が Haskell で実装されていることから、express の実装には、Haskell の Web アプリケーションフレームワーク Yesod [15] が用いられている。本論文では express の改良を目標とする。

3.2 GMB Explorer

GMB (Groningen Meaning Bank) [16] は、パブリックドメインの英語テキストからなる統語・意味情報付きの大規模コーパスで、統語理論に CCG を、意味計算に談話表示理論 (Discourse Representation Theory; DRT) [17] を採用している。GMB は現在テキスト形式でのダウンロードが可能であるが、テキスト形式では、品詞や意味の表示、あるいは構造の特性についての情報を十分に把握することが困難である。そこで GMB Explorer という Web インターフェースを用いることで、ユーザーは即座に文の統語構造をわかるように表示された GMB にアクセスすることができる。例えば、機能として統語構造一部の折り畳み/展開機能があり、この機能によって、ユーザーは他の構造を折り畳んだまま、特定の部分構造だけを展開して表示できるため、分析対象の部分構造に迅速かつ効率的にアクセスすることができる。

3.3 PMB Explorer

PMB (Parallel Meaning Bank) [18] は、GMB を基盤にした統語・意味情報付きの多言語コーパスである。PMB も GMB 同様、PMB Explorer という Web インタフェースを用いることで、ユーザは視認性に優れた PMB にアクセスすることができる。さらに、GMB Explorer の機能に加えて、専門家による手動修正が可能である。この機能によって、専門家がコーパスの解析結果に間違いを見つけた際に即座にインタフェース上で間違いを修正することができるため、より効率的に分析や、修正を行うことができる。

4 提案手法

4.1 パイプライン

express には以下の 3 つの画面（付録図 8 参照）がある。図 2 に express の実装パイプラインを示す。

JSeM JSeM を用いた NLI 結果の可視化

Parsing 入力文の統語解析、型検査結果の可視化

ChartParsing チャートパーズングの計算過程の可視化
メニュー画面を用意したことによって分析

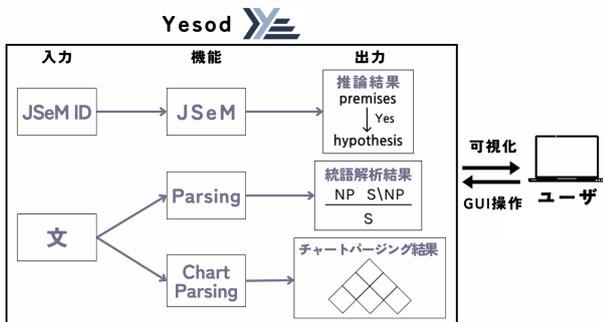


図 2 express のパイプライン

対象の解析結果に直接アクセスすることが可能になった。詳しい機能については 4.2 節で説明する。

4.2 express の機能

本研究で行った主な実装のうち、5 つの機能・特徴をそれぞれ説明する。

4.2.1 入力文の統語・意味情報の可視化

入力文に対して、CCG 統語構造・DTS 意味表示の可視化を行えるよう実装した。加えて、出力された統語構造の一部の折り畳み／展開機能も実装した（図 3 参照）。統語構造の一部の折り畳み／展開機能

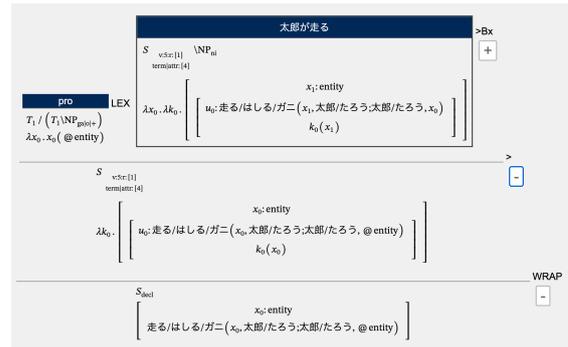


図 3 「太郎が走る」の解析結果の一部を折り畳んだ構造

はそれぞれボタンと+ボタンで行っている。初期状態は親ノード以外全て畳まれている状態が表示され、+ボタンを押すとその一つ下の子ノードが展開される。-ボタンを押すと、その子ノードが折り畳まれる。

ユーザは express にアクセスすると、入力文と beam 数（解析結果をいくつ使うかを表した数）をフォーム形式で指定することで入力文の統語構造と意味表示を beam 数だけ得ることができる。この実装により、展開したい部分構造を、他の構造を折りたたんだままの状態を展開できるため、より効率的に調査したい部分構造に到達することができるようになった。

4.2.2 チャートパーズングの可視化

lightblue では、統語解析時に、文に対して左隅チャートパーズング (left-corner chart parsing) [19] を行う。左隅チャートパーズングでは、左下から解析を始め、部分文字列すべての可能な組み合わせについてスコアを求め、スコアの高い組み合わせをビーム数だけ出力する。それゆえ、lightblue は複数の統語解析結果を計算できるという特徴を持つ。

そこで本手法ではこの左隅チャートパーズング図の可視化を実装した（図 4 参照）。マスにカーソルを当てると、マスが黄色で塗られる仕様になっている。そして、それぞれのマスをクリックすることで、各マスの CCG 統語範疇・DTS 意味表示が図 3 のように表示される。この機能によって、統語解析が失敗したときに、デバック効率が上がり、原因特定がしやすくなる。

4.2.3 型検査証明図の可視化

入力文の意味表示に対応する型検査クエリの型検査証明図を表示する機能を実装した。型検査クエ

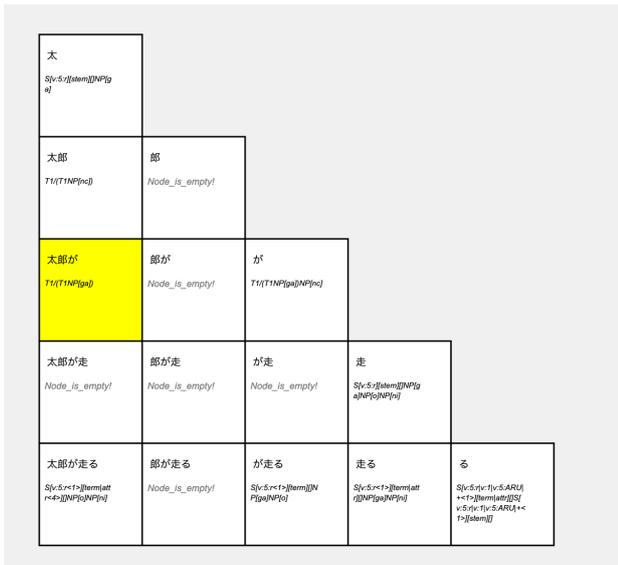


図4 「太郎が走る」のチャートパーズング図の可視化

リ (Type Check Query) とは、証明の対象となる文の意味表示が type であるかという問いのことである。型検査証明図の実装によって、図5のように証明図の一部を折り畳み/展開することが可能となり、型検査の詳細なプロセスが、視覚的に明瞭になり、全体像を素早く把握できるように改善された。

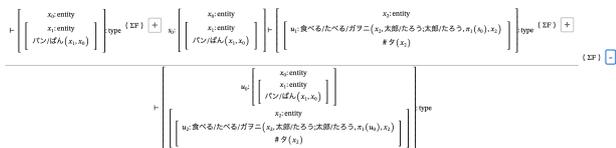


図5 「太郎がパンを食べた」の型検査証明図の一部を折り畳んだ証明図

4.2.4 型検査失敗時の表示

型検査は、証明ができない場合は失敗とみなされる。本研究では、入力文の型検査に失敗したとき、証明が失敗した型検査クエリを特定しやすくするインターフェースを実装した。この実装結果を図6に示す。

lightblue の型検査クエリ (図6の下部の Type Check Query) は統語解析結果の数だけ存在し、型検査は型検査クエリそれぞれに対して行われる。そこで、複数の型検査クエリをタブ形式で切り替えて確認する機能を追加した。加えて、型検査に失敗している、つまり型検査の証明図 (Type Check Diagram) が見つからなかった型検査クエリに対しては、タブの色を赤色にするよう実装した。この実装により、型検査に失敗しているときにそれがどの型検査クエリによるものなのかを特定しやすくなり、デバッグの

効率化が見込まれる。

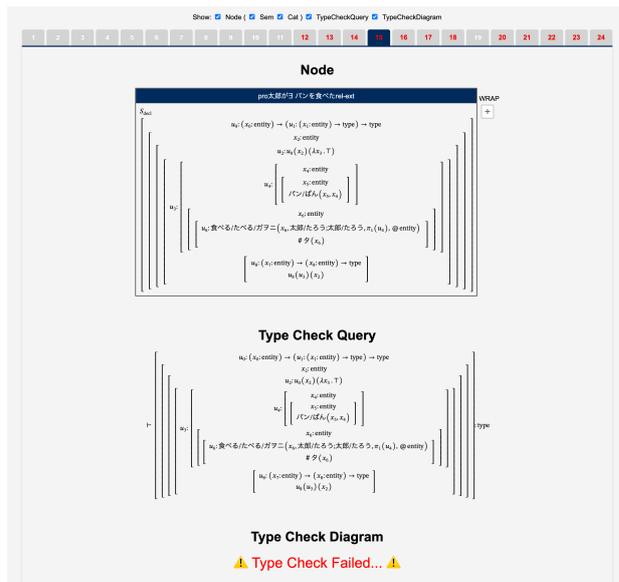


図6 Parsing 画面の解析結果表示

4.2.5 推論証明図の可視化

石嶋ら [10] は、JSeM の各データに割り当てられた ID を入力するとその JSeM データの前提文と仮説文の統語構造を可視化する機能を実装した。本手法では、前提文と仮説文の統語構造の可視化に加え、推論結果の証明図の表示機能の実装を実現した。

この実装では、推論結果の証明図は、型検査証明図と同様に一部を折り畳み/展開することができる。これによって、一目で推論証明図の構造がわかるようになり、証明の流れや文法規則が直感的に理解しやすくなった。推論証明図の可視化結果は付録図9に示す。

5 まとめ・今後の課題

Web アプリケーションフレームワーク Yesod を用いて、GUI を備えた lightblue 開発システム、express を改良した。本システムを用いることで、チャートパーズング過程や型検査、自然言語推論の証明図の可視化が可能になった。加えて、文法開発に関わるユーザの解析効率や解析結果の視認性を上げるような、証明図一部の折り畳み/展開機能や、型検査失敗時にはそのことを表示をする機能を実装した。

今後は、文を解析して出力された CCG ツリーバンクの保存機能や PMB Explorer のような解析結果の編集機能を新たに実装することで文法開発の進展に貢献していく。

謝辞 本研究の一部は、JSPS 科研費 JP23H03452 の助成を受けたものである。

参考文献

- [1] Koji Mineshima, Pascual Martínez-Gómez, Yusuke Miyao, and Daisuke Bekki. Higher-order logical inference with compositional semantics. In **Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing**, pp. 2055–2061, Lisbon, Portugal, September 2015. Association for Computational Linguistics.
- [2] Daisuke Bekki and Ai Kawazoe. Implementing variable vectors in a CCG parser. In **Logical Aspects of Computational Linguistics. Celebrating 20 Years of LACL (1996–2016)**, pp. 52–67, Berlin, Heidelberg, 12 2016. Springer Berlin Heidelberg.
- [3] Daisuke Bekki and Koji Mineshima. Context-passing and underspecification in dependent type semantics. In Stergios Chatzikyriakidis and Zhaohui Luo, editors, **Studies of Linguistics and Philosophy**, pp. 11–41. Springer International Publishing, 2017.
- [4] Per Martin-Löf. **Intuitionistic Type Theory**, Vol. 17. Bibliopolis, 1984.
- [5] Hinari Daido. DTS の部分体系を用いた定理自動証明器への等号型の導入. PhD thesis, お茶の水女子大学, 2022.
- [6] Hinari Daido and Daisuke Bekki. Development of an automated theorem prover for the fragment of dts. In **the 17th International Workshop on Logic and Engineering of Natural Language Semantics (LENLS17)**, 2020.
- [7] 富田朝, 谷中瞳, 戸次大介. 言語学的に妥当な日本語 CCG ツリーバンクの構築と評価. 言語処理学会第 30 回年次大会 (NLP2024), 2024.
- [8] Asa Tomita, Hitomi Yanaka, and Daisuke Bekki. Reforging : A method for constructing a linguistically valid Japanese CCG treebank. In Neele Falk, Sara Papi, and Mike Zhang, editors, **Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop**, pp. 196–207, St. Julian’s, Malta, March 2024. Association for Computational Linguistics.
- [9] 松原舞, 富田朝, 戸次大介. CCG 統語解析器 lightblue と定理証明器 wani による JSeM Verbs データセットの自動推論. 言語処理学会第 31 回年次大会 (NLP2025), 2025.
- [10] 石嶋美咲, 戸次大介. Yesod による日本語 CCG パーザ開発環境の構築. 人工知能学会全国大会論文集 第 35 回全国大会 (2021), 2021.
- [11] Mark Steedman. **Surface Structure and Interpretation**. The MIT Press, Cambridge, 1996.
- [12] Mark Steedman. **The Syntactic Process**. MIT Press, 2000.
- [13] Daisuke Bekki and Miho Satoh. Calculating projections via type checking. In **Proceedings of the TYTTLES Workshop on Type Theory and Lexical Semantics, ESSLLI**, 2015.
- [14] Ai Kawazoe, Ribeka Tanaka, Koji Mineshima, and Daisuke Bekki. An inference problem set for evaluating semantic theories and semantic processing systems for Japanese. In **the 17th International Workshop on Logic and Engineering of Natural Language Semantics (LENLS12)**, pp. 67–73, 2015.
- [15] Michael Snoyman. **Developing Web Apps with Haskell and Yesod**, Vol. 2. O’Reilly Media, 2015.
- [16] Valerio Basile, Johan Bos, Kilian Evang, and Noortje Venhuizen. A platform for collaborative semantic annotation. In **Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL)**, 2012.
- [17] Hans Kamp. A theory of truth and semantic representation. In **Formal Methods in the Study of Language**, pp. 277–322, 1981.
- [18] Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. The Parallel Meaning Bank: Towards a Multilingual Corpus of Translations Annotated with Compositional Meaning Representations. In **Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)**, pp. 242–247, 2017.
- [19] Klaas Sikkel. **Left-Corner chart parsing**, pp. 201–220. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

A 付録

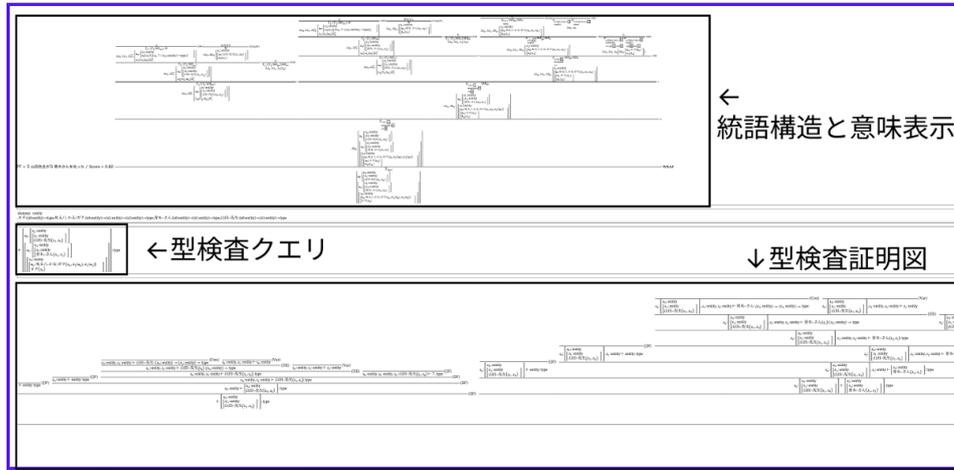


図 7 可視化前の lightblue による HTML 出力のキャプチャ画面

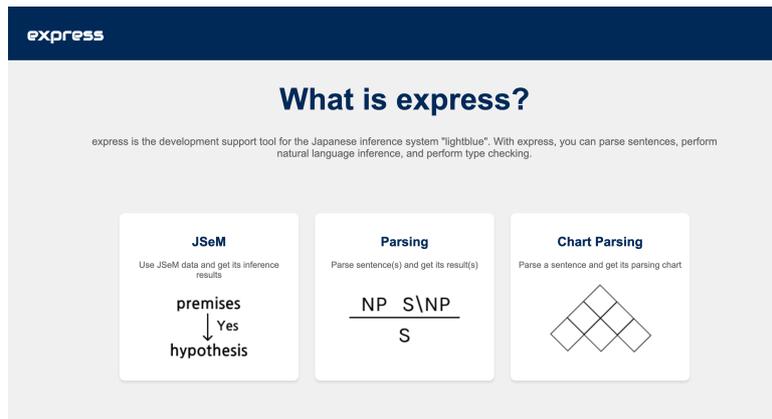


図 8 express のメニュー画面

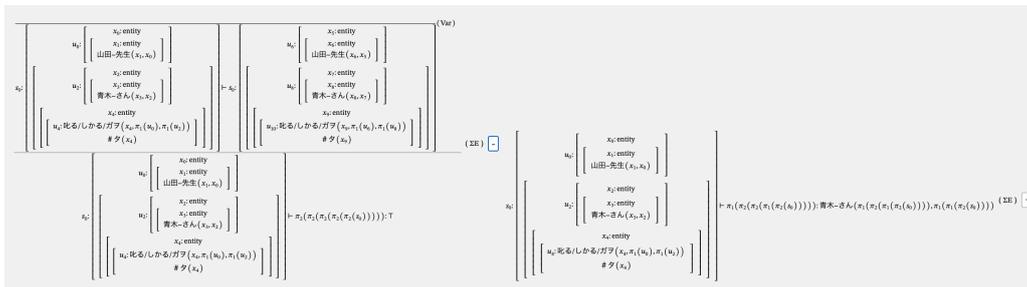


図 9 前提文「山田先生が青木さんを叱った。」と仮説文「青木さんが山田先生に叱られた。」における推論証明図の一部を折り畳んだ構造