Q6-13

Yesod による日本語推論システムlightblue の開発環境の構築

佐伯小遥　富田朝　松原舞　戸次大介
お茶の水女子大学
{saeki.koharu, tomita.asa, matsubara.mai, bekki}@is.ocha.ac.jp

はじめに：日本語自動推論システム lightblue による推論パイプライン

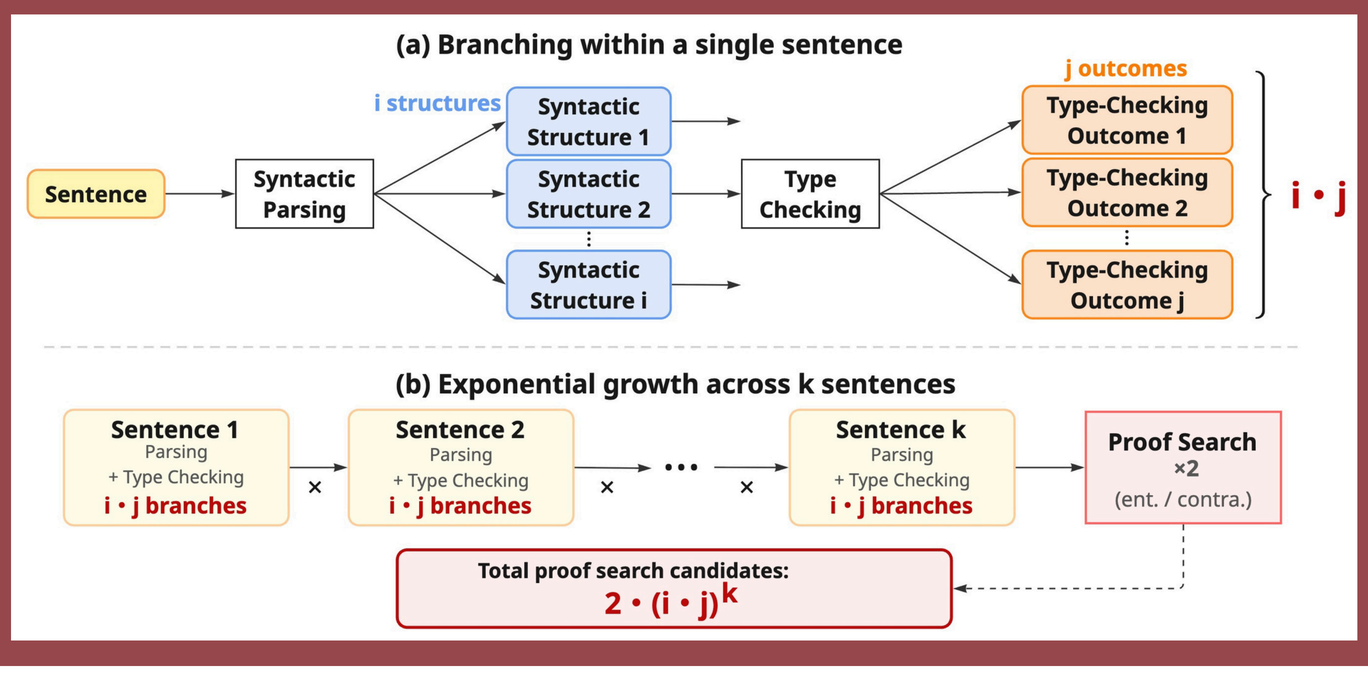
lightblueによる推論は統語/意味解析，型検査，証明探索の3段階から構成され，各段階で理論的に許容される複数の候補を保持しながら進行するため以下の問題がある．

問題：分岐の指数的増大

- 従来：候補をすべて列挙した上で推論を進める **全列挙型実行方式**
- k 個の入力文に対し，各文で i 通りの統語構造と j 通りの型検査結果が生じるため，証明探索の候補数は最大 $2\cdot(i\cdot j)^k$ に達する（右図参照）．

① 不要な経路も一律に計算 → **計算コスト**大
② 様々な推論経路が混在した出力 → 特定分岐の**検証が困難**

修正を加えた後にそれが統語構造に反映されているかを検証したい…

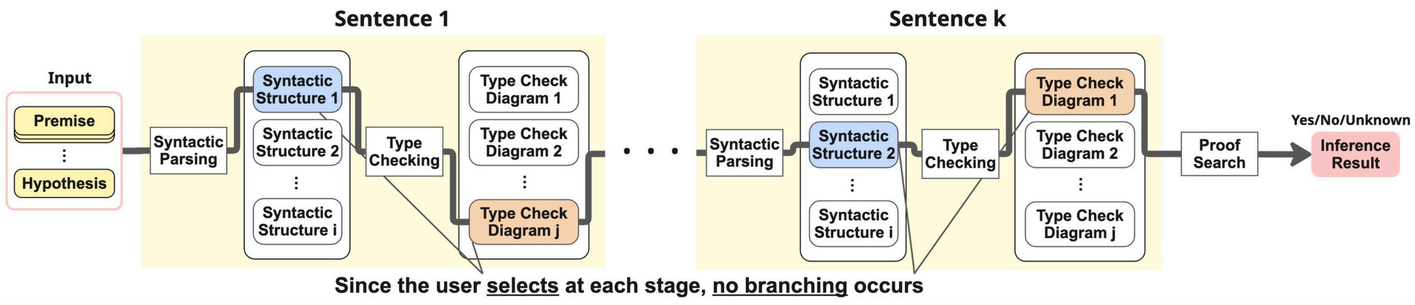


提案手法：Express による推論過程の対話的再構成

lightblue における推論を，一括実行される処理としてではなく，分岐を伴う探索過程として捉え直し，その過程を文法開発者が段階的/選択的に操作可能な **対話的プロセス** として再定義する開発環境 **Express** を提案する．

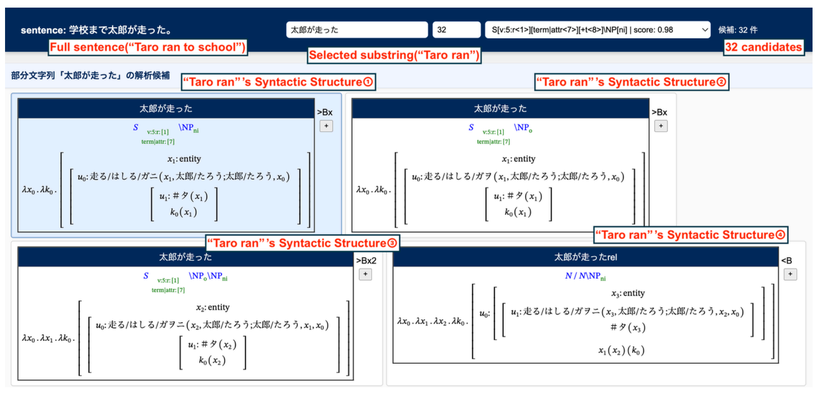
推論過程に内在する分岐点そのものを明示化し，それを開発者が操作する単位として定義した（右下図参照）：
(1) 統語解析段階における統語構造の選択：CCGに基づく解析で得られる複数の統語構造を並列に提示し，開発者は注目する構造を明示的に選択する．
(2) 型検査段階における型検査結果の選択：照応解決により生じる複数の型検査証明図を個別の分岐として提示し，文脈としてどの結果を採用するかを指定する．

文法開発の典型的な場面として，推論結果が期待するラベルと異なる場合を考える．このとき文法開発者は，**(a) 問題の局所化**，**(b) 修正の全体検証**，**(c) 失敗条件の共有** という段階を経て原因の特定と修正を行う．
Express はこの各段階を以下の3機能により直接支援する．



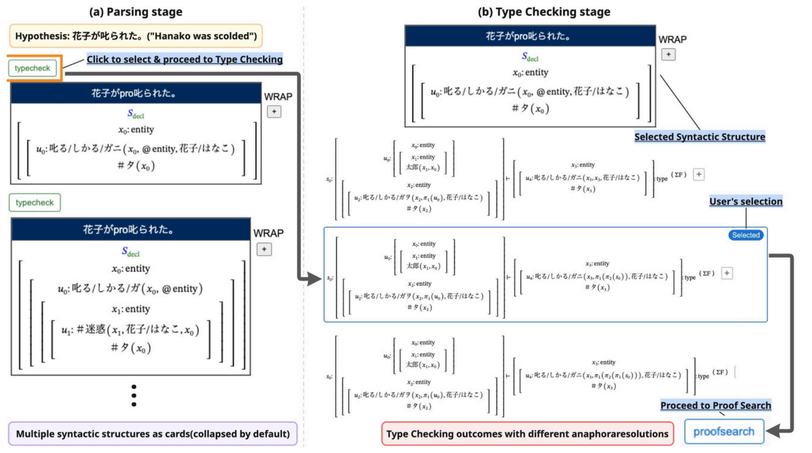
(a) 部分文字列解析機能

パイプライン全体の再実行をせずに，入力文のうちドラッグ選択した部分文字列に対する全統語構造に即座にアクセスできる．



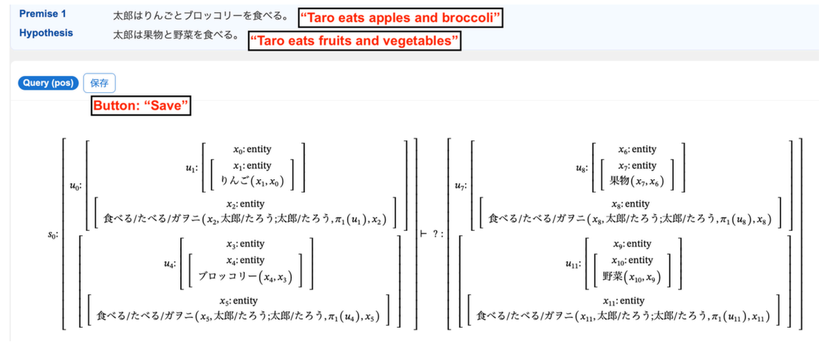
(b) 対話的ブランチ選択機能

複数の統語構造と型検査結果を開発者が各段階で1つずつ選択しながら推論を進める．



(c) 証明探索クエリのエクスポート機能

対話的に選択した推論経路に対応する証明探索クエリを，lightblue の定理証明器 wani が直接読み込み可能な形式でエクスポートできる．



評価実験

(1) 探索空間の定量的評価

ユーザスタディで使用した2問を用い，全列挙型とExpressにおける探索パス数および計算時間を比較した．全列挙型はいずれの問題でも120秒の制限時間内に全探索を完了できなかった．Express では，証明探索クエリをエクスポート可能であり，証明探索が完了しなくても，全列挙型実行の終了を**待つことなく，問題の特定と共有に着手できる**．

	Exhaustive		Express	
	Paths	Time	Paths	Time
Problem A	260	>120s	10	12s
Problem B	128	>120s	28	94s

(2) ユーザスタディ

CCG・DTS等の専門知識を持つlightblueの開発者4名を対象にwithin-subjects デザインによるユーザスタディを実施した．指定された推論経路を同定し，対応する証明探索クエリを取得するというタスクを行なった．Baseline 条件では4名中3名が10分の制限時間内にタスクを完了できなかった．唯一成功した参加者の場合でも，**Express 使用時と比較して約3倍の時間**を要した．

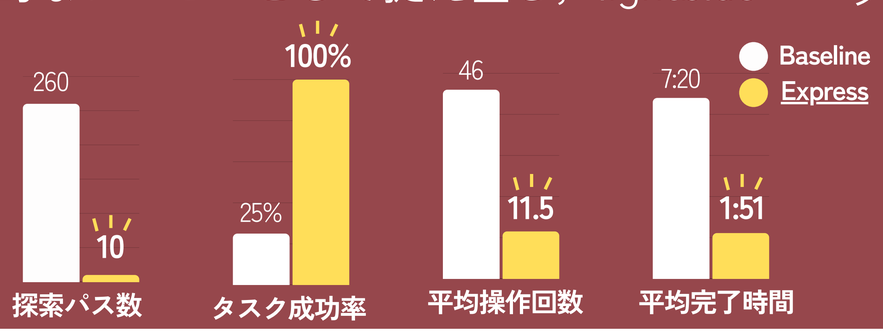
	成功率	完了時間	操作回数
Baseline	25% (1/4)	7m 20s	46
Express	**100% (4/4)**	**1m 51s**	**11.5**

(3) ケーススタディ

期待するラベルが YES であるのに lightblue が UNKNOWN を返す問題について検証を行った．Express により対話的に推論経路を絞り込み，問題となる証明探索クエリを特定・エクスポートした．エクスポートしたファイルを wani 開発者に共有したところ，約5分で根本原因が特定された．従来はスクリーンショットからDTT 形式への手動転記に約1時間を要しており，**約12倍の短縮**となった．根本原因の特定後，文法開発者が語彙項目を追加・再コンパイルし，Express 上で修正後の推論結果が期待通り YES となることを確認した．これにより，調査→修正→検証のサイクルが実現された．

まとめ

- 推論過程を文法開発者との相互作用を通じて実行される対話的なプロセスとして捉え直し，lightblue の推論を**対話的実行可能**にする **Express** を構築した．
- 従来の全列挙型実行方式と比較し，以下の結果を達成した．
  - ☑ 探索パス **最大96%削減**
  - ☑ タスク成功率 **25% → 100%**
  - ☑ 平均操作回数 **約4倍削減**
  - ☑ デバッグ時間 **約12倍短縮**



ライブデモはこちら