学会記事

Non Peer-Reviewed

記号処理への回帰：パターンに基づく速度指向言語処理

吉永　直樹 †

1　はじめに

本稿では，ACL2023 に採択された論文 “Back to Patterns: Efficient Japanese Morphological Analysis with Feature-Sequence Trie” (Yoshinaga 2023) について，解説する．本研究は，処理の高速性，省メモリ性，精度を兼ね備えた理想的な基礎解析を実現すべく，古典的な記号処理に基づく言語処理を再訪して，日本語の形態素解析においてその可能性を探求したものである．研究室主催者でもある著者が，短期間（約 3 ヶ月）で，手法の考案・実装，実験，論文の執筆を行った研究であり，深層学習に基づく手法が主流の時代において機械学習すら用いていないなど，振り切った視点で行った研究でもある．この点を踏まえ，研究の経緯を中心に説明する．なお，手法の実装である Jagger は下記のサイト[1]で公開している．

2　提案手法の概要

提案手法は，形態素解析（単語分割・品詞タグ付け・見出し語化の混合タスク）を，単語分割位置と切り出した単語の品詞・見出し語を順次同定する超多クラス分類問題とみなし，これを機械学習で有効となる特徴量（後文脈表層，前文脈品詞）を並べたパターンを用いて解くことで超高速に形態素解析を行う．特徴量（パターン）に対する分類結果，すなわち分割位置・品詞・見出し語は，そのパターンに対し学習データ中で最頻となる分割位置・品詞・見出し語を割り当てる．本手法では，最長一致する特徴パターンを入力の先頭から順次適用するが，分類結果が同じ冗長なパターンを枝刈りすることで，処理速度と省メモリ性が維持されている（図 1）．

提案手法の説明は以上であり，その実装は C++で 1000 行弱であるなど簡潔であるが，標準コーパスを用いた実験では，最小コスト法 (Kudo et al. 2004) を実装した MeCab[2] (Vibrato[3]) や点推定 (Neubig et al. 2011) を実装した Vaporetto[4]など既存の形態素解析手法の効率の良い実装に対し，約 7–16 倍の処理速度，1/2–1/20 の消費メモリで，遜色のない精度の解析を行うことができる．以降，本手法を考案するに至った経緯について，時系列を追って説明する．

† 東京大学生産技術研究所

1 http://www.tkl.iis.u-tokyo.ac.jp/~ynaga/jagger/

2 https://taku910.github.io/mecab/

3 https://github.com/daac-tools/vibrato

4 https://github.com/daac-tools/vaporetto

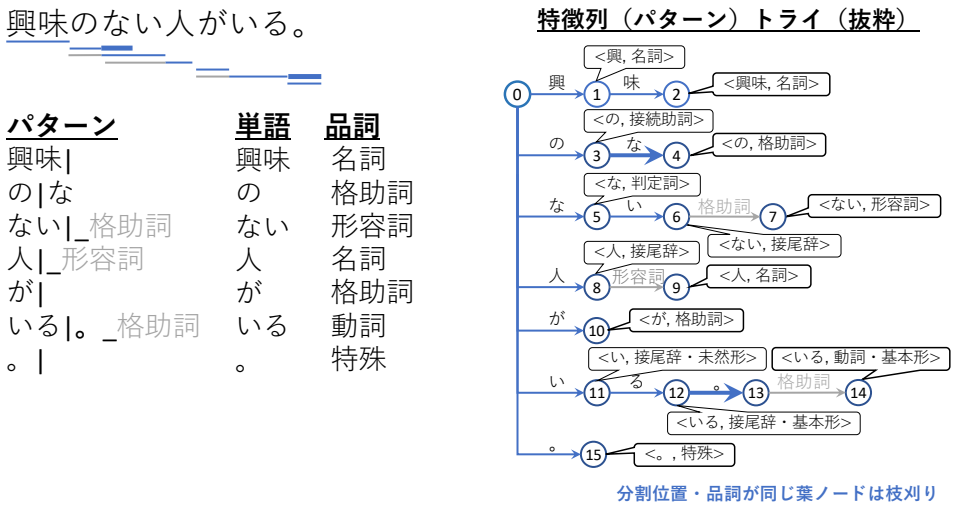

図 1　最長一致パターンに基づく形態素解析：助詞「の」は接続助詞または格助詞になり得るが，形容詞「ない」が後続することから格助詞と判断できる．また，「のない」に対するパターンは，対応する単語分割位置・品詞が「のな」と同じであることから枝刈りされており，冗長な後文脈表層を読み込むことなく直ちに処理を確定できている．「ない」については，接尾辞の可能性もあるが，前文脈品詞が格助詞のときは形容詞が学習コーパス中で最頻となることから，正しく解析が行えている．

3　本研究の経緯

自然言語処理分野では，過去 30 年に亘ってベンチマークデータセットにおける解析精度を主目標として研究開発が行われているが，そのような精度に偏重した技術は，実際に技術を利用する場面において必ずしも求められていない．膨大なソーシャルメディアテキストを分析対象とする社会情報学や，無数のユーザクエリを処理する E コマースなどでは，精度（出力の質）と速度（運用コスト）のトレードオフを考慮して用いる技術が決定される．特に，最も多用される基礎解析では，高精度でも非効率な手法を採用できる環境は少ない．著者自身も，ブログ投稿や X（旧 Twitter）の投稿など膨大な実世界テキストを対象とした社会分析システムを実装する過程でこの問題を意識するに至り，形態素解析器 (MeCab) と同程度の速度で動作する係り受け解析器 J.DepP[5] (Yoshinaga and Kitsuregawa 2009, 2014) を実装するなどした．

前述の社会分析システムでは，形態素解析器として MeCab を利用していたが，コロナ禍に入り全量 X（旧 Twitter）投稿を超リアルタイムで解析する社会分析システム[6]を構築することになり，形態素解析の処理速度がボトルネックとなるようになった．形態素解析については，

[5] http://www.tkl.iis.u-tokyo.ac.jp/~ynaga/jdepp/
[6] 数ヶ月〜数年分の投稿を数分〜数十分程度で解析する．全量 X 投稿は膨大で解析結果をディスクに保存することが難しく，辞書等でモデルを微調整して再解析を行う機会も多いため，オンラインで解析する実装となっている．

MeCab（旧 ChaSenTNG）以降 20 年近くより効率に優れた実装が公開されることはなく，処理速度の観点では技術的に長く停滞[7]していた．2021 年になって，点推定に基づく形態素解析器の効率を改善した Vaporetto が公開されたものの，公開時のバージョンは単語分割速度に焦点を当てたもので，品詞タグ付けまで行うと遅く，MeCab を置き換えられるものではなかった．

このように速度面での言語処理技術の停滞（正確には，後退）にモヤモヤしていたところで，“Efficient Methods for Natural Language Processing: A Survey”(Treviso et al. 2023) を読む機会があり，紹介されている「効率的」な手法が，基本的に超低速な深層学習手法を限定的に高速化するマッチポンプな高速化手法であることに失望した．現在の自然言語処理研究者は，その多くが「精度至上主義の呪い」に囚われており，自分が必要とする速度指向の言語処理技術は自分で作るしかないと観念するに至り，最速で実用的な精度の形態素解析器を作ることにした．

3.1 手法の考案・実装

EMNLP2022 のリバッタルへの対応が終わった 2022 年 9 月 5 日に，実装するアルゴリズムの検討を行った．深層学習の利用については，技術的に最速の処理を実現することが困難なことやその実験コストなどから（学生の実験を邪魔することを嫌って）考えなかったが，機械学習の利用までは排除しておらず，単語分割では，最速の最長一致法をもとに必要に応じて分類器を併用することで，処理速度を維持して解析精度を高める方針[8]を立てた．また，品詞タグ付け（見出し語化）については，ラベル数に比例して遅くなる分類器の argmax 計算の高速化に焦点を当てていた（研究メモによれば，コーパス中で最頻の品詞かどうか分類し，最頻でないときのみ多値分類するアプローチを検討していた）．これら基本方針を立てたのち，業務の隙間時間を利用して，短期集中的に手法の実装と改良を進めた．

9 月 10 日の昼から，自作の動的ダブル配列 cedar (Yoshinaga and Kitsuregawa 2014) を用いて最長一致法を用いた単語分割器を実装し，MeCab の約 6 倍の処理速度が出ることを確認した．開発データの解析結果で助詞を後続の平仮名と連結する誤りが多く見られたので，分割位置の後方文脈の表層まで追加で見て最頻の分割位置で分割する手法を実装したところ，処理速度を維持して最小コスト法 (MeCab) や点推定 (Vaporetto) と遜色ない精度で単語分割を行うことができることが分かった．これらの実装に要した時間は約 9 時間で，単語分割器は C++で 62 行，パターンの抽出は Python で 45 行と極めて簡素なものであった．このように簡素な手法で機械学習を用いた手法と遜色のない単語分割精度が得られたことは驚きであり，パターンに基づく記号処理の可能性を強く意識することとなった．

[7] 自然言語処理という閉じた学術分野では，形態素解析は，構文解析などのより深い基礎解析や機械翻訳などの応用を実現するための前処理として位置付けられているが，ベンチマークデータにおける解析精度が飽和し，深層学習の導入で前処理として形態素解析を用いる機会が減ったことで，研究自体がほとんど行われなくなってきている．

[8] 同様の方針で，単語分割において・最長一致法の速度を維持して解析精度を改善した手法として (Sassano 2014) があるが，機械学習に基づく手法 (MeCab) とは 1%以上の精度差があったため，この差を埋めることを目標とした．

9 月 11 日の夜に，最長一致する後文脈表層をパターンとする提案手法を機械学習的な観点で再考し「パターンは，後文脈のどの位置で切るかという多値分類問題の解を事前計算したものである」という発想に至り，自身が過去に行った「非線形分類器において，実際の言語データに出現する事例について分類結果を事前計算しておき，入力の近傍事例の分類結果を参照して分類を高速化」する研究 (Yoshinaga and Kitsuregawa 2009, 2014) との類似性にも気がついた．点推定では，各文字の後で単語が分割されるか，という二値分類問題を解くが，提案手法では，次にどの位置で分割されるか，という多値分類問題を解くことで，分類回数を削減している．さらに機械学習の分類器（スコアの argmax 計算）を経由せず，パターンを用いて直接分類結果（分割位置）を取得する点で，アルゴリズム的により高速なものとなっている．このアイデアの素直な拡張として，分割位置に加えて切り出した単語の品詞を同時に推定する多値分類問題を考え，パターン（特徴量）に対して最頻の分割位置・品詞を学習データから計算して割り当てれば，品詞タグ付けまで高精度で解けるかもしれない，という感触を得た．

9 月 20 日に 2 時間ほど作業して上記のアイデアを追加実装し，品詞タグ付けまで行うようにしたところ，MeCab に対して処理速度は 4 倍となったが，解析精度は 1.8%劣るものとなった．そこで，パターンに追加で特徴量を追加して精度の改善を行うこととした．9 月 25 日に単純なオンライン学習を用いて点推定で品詞タグ付けのみ行う分類器を実装して有効となる特徴量を調べ，前文脈品詞が特に有効であることが分かった．その後しばらく，実装が複雑になることを嫌って（まとまった時間も取れず）放置していたが，EACL2023 への概要投稿後，10 月 14〜16 日にかけて実装し，MeCab や Vaporetto と遜色ない解析精度を得るに至った．品詞タグ付け（見出し語化）結果を出力する場合，単語分割結果のみ出力する場合と比べて処理速度が大きく低下する問題があったが，出力の成形で行っていた文字列終端 '\0' を辿る操作がボトルネックであることを突き止め，論文締め切りの数日前，10 月 18 日には京都大学テキストコーパスを M2 MacBook Air で約 50 万文/秒で解析する形態素解析器を実現することができた．

3.2 論文投稿から採択まで

成果をまとめるに当たり，1 月締め切りの ACL2023 を最終的な出版先として想定したものの，深層学習どころか機械学習すら用いない記号処理に基づく基礎解析手法が，昨今の学会で関心を持ってもらえるかは未知数であった．また，学会では近年，手法の多言語への汎用性が重視されており，日本語のみを対象とする評価実験で成果が認められるかも不透明であった．そこで，研究コミュニティの反応を見るため（本会議に採択されたら出版するつもりで）EACL2023（10 月 21 日夜 9 時締切）に論文を投稿することにした．EACL2023 には，学生との共著論文も複数，投稿することとなったため，（共著論文への対応が一段落した）締切当日未明から大急ぎで論文を執筆し，午前に実験を行なって，午後早い時間までに投稿を済ませた．

EACL2023 投稿後は，近年の学会の価値観に合わせて，多言語での評価や深層学習モデルの

近似などで高精度化を目指すことも検討したが，実際に自分が必要としない研究のためだけの実装や実験をする気にはなれなかったため，手法の長所である処理速度を限界まで改善すべく，隙間時間に実装の細かい最適化を続けた．具体的な目標として，京都大学テキストコーパスで 100 万文/秒を設定したが，果たして 50 万文/秒から 100 万文/秒までの道のりは，険しかった（同時に本研究で最も楽しかったのはこの期間であった）．J.DepP の開発時に培った最適化技法は既に投入しており，打てる手はないと何度も諦めかけたが，高速化のためのアイデアを実装し続け，最終的に 12 月 7 日に 100 万文/秒を達成した．Vibrato でも用いられている Unicode の文字単位で遷移するダブル配列（11 月 18 日実装）や出力のバッファリング（11 月 24 日実装）が特に有効であったが，ビット単位のデータ管理など細かい工夫の積み重ね無しに 100 万文/秒は達成できなかっただろう．一方で，効果を期待した戻り読みのないパターンマッチや SIMD を用いた UTF8 の Unicode のコードポイントへの変換などは，逆効果であり，苦い思いをした．

100 万文/秒を達成した直後に届いた EACL2023 の査読のスコアは 4/4/3 であり，評価が渋いショート論文であることを考慮すると本会議での採択も期待できた．100 万文/秒という結果を持って ACL2023 に挑戦したい気持ちも出てきたところで，悩ましい気持ちで採否通知を待っていたが，ACL2023 投稿直後に来た EACL2023 の採否通知で，残念ながら（結果的には運良く）Findings に回されたため，withdraw して ACL2023 の採否を待つこととした．ACL2023 では，最終的に soundness 4/4/4, excitement 4.5/3.5/4.5 と良いスコアで本会議に採択された．

4 研究を振り返って

研究室を主催する大学教員の多くは，学内外の業務や教育研究に忙殺され，少しずつ自身で手を動かす現役の研究者ではなくなっていく．かく言う自分も研究室を主宰するようになって，自分で手を動かす研究からは離れていた．

しかし，やはり研究は自分で自由に進めてこそ，楽しいものだ．自分の場合，伴走者の立場で学生の研究に関わる時間が長くなるにつれて，少しずつ，学生が好む学会で流行する研究と自身の興味のある研究との間に乖離を感じるようになっていた．また，研究を主体的に進める当事者感覚が失われていく怖さもあった．本研究は，これらを解消すべく進めたものである．研究時間を捻出するのは難しかったが，過去に培った研究経験のおかげで，研究で最も楽しい（実装を磨く）時間に多くの時間を費やすことができ，深層学習の呪いから自分を解放することもできた．この研究に触発され，再び主体的に研究に従事する大学教員が増えると嬉しい．

自然言語処理分野は，しばらく，数字の引用で既存研究に対する優位性を示せる解析精度という単一の評価指標に集中することで技術を発展させてきた．その中で，処理速度については，テキストデータの量が増え続けているにもかかわらず，計算資源の発達に甘えて（お金で解決できる問題とみなし）軽視されてきた．その結果，「頭でっかち」のモデルが世に溢れ，計算機

を用いた処理の長所であった処理速度は損なわれ続けている．近年の大規模言語モデルの氾濫で，今や解析精度もお金（学習データを増やして大規模な学習を行うこと）で解決できる問題とみなされつつあることに皮肉を感じる．個人的には，自然言語処理は，最速・最小・最高精度のモデルを追求すべきであり，その追求が言語のかたちを明らかにすることにも繋がると考えている．この点では，本研究も，たどり来て，未だ山麓という印象である（解析精度が不十分）．

本研究で再訪した記号処理には，深層学習モデルの離散化（近似）や記号推論との接続など，様々な展開が期待できる．記号処理では，深層学習モデルに特有の効率や解釈性の問題はなく，仮説やアイデアを検証する実験が終わるのを，何日も待つ必要もない（実験は，数秒で終わる）．本研究に続き，学会の流行に流されず，自身の信念に従って行われる研究が増えることを願う．

参考文献

Kudo, T., Yamamoto, K., and Matsumoto, Y. (2004). “Applying Conditional Random Fields to Japanese Morphological Analysis.” In *Proceedings of EMNLP*, pp. 230–237.

Neubig, G., Nakata, Y., and Mori, S. (2011). “Pointwise Prediction for Robust, Adaptable Japanese Morphological Analysis.” In *Proceedings of ACL: HLT*, pp. 529–533.

Sassano, M. (2014). “Deterministic Word Segmentation Using Maximum Matching with Fully Lexicalized Rules.” In *Proceedings of EACL (Volume 2)*, pp. 79–83.

Treviso, M., Lee, J.-U., Ji, T., Aken, B. v., Cao, Q., Ciosici, M. R., Hassid, M., Heafield, K., Hooker, S., Raffel, C., Martins, P. H., Martins, A. F. T., Forde, J. Z., Milder, P., Simpson, E., Slonim, N., Dodge, J., Strubell, E., Balasubramanian, N., Derczynski, L., Gurevych, I., and Schwartz, R. (2023). “Efficient Methods for Natural Language Processing: A Survey.” *TACL*, **11**, pp. 826–860.

Yoshinaga, N. (2023). “Back to Patterns: Efficient Japanese Morphological Analysis with Feature-Sequence Trie.” In *Proceedings of ACL (Volume 2)*, pp. 13–23.

Yoshinaga, N. and Kitsuregawa, M. (2009). “Polynomial to Linear: Efficient Classification with Conjunctive Features.” In *Proceedings of EMNLP*, pp. 1542–1551.

Yoshinaga, N. and Kitsuregawa, M. (2014). “A Self-adaptive Classifier for Efficient Text-stream Processing.” In *Proceedings of COLING*, pp. 1091–1102.

略歴

吉永　直樹：2005 年東京大学大学院情報理工学系研究科博士後期課程修了．博士（情報理工学）．2016 年より東京大学生産技術研究所准教授．主に，超高速な基礎解析，即時的な知識獲得，適応的な言語生成の研究に従事．