確率木置換文法と分散表現に基づく レシピ構造木の生成モデルの学習

吉成 未菜里 [1]　横井 祥 [1,3]　進藤 裕之 [2,3]　乾 健太郎 [1,3]
[1] 東北大学　[2] 奈良先端科学技術大学院大学　[3] 理化学研究所 AIP センター
{yoshinari, yokoi, inui}@ecei.tohoku.ac.jp　shindo@is.naist.jp

1 はじめに

一般ユーザが CGM (Consumer Generated Media)[*1] に投稿した大量の料理レシピを利活用するため，検索，分類，生成といった料理レシピの情報処理の研究が活発におこなわれている [4, 13, 15][*2]．

各レシピの意味表現としては，特に，**調理手順を木やグラフによって構造化**する試みが盛んである．例えば Mori らはレシピの無閉路有向グラフによる表現 *Flow Graph* を定義し，日本語のレシピ投稿サイトであるクックパッド[*3]のレシピの一部を Flow Graph 形式に変換したコーパスを作成した [9]．また Wang らは調理動作の順序を表す *action flow* と材料に対して適用される動作を表す *ingredient flow* の 2 種類のフローを持つレシピのグラフ表現 *Cooking Graph* を定義し [14]，これを中国語のレシピ検索[*4]に活用した [15]．以下本稿ではレシピの表現としてとくに木構造を扱う（実際に実験に用いたレシピの木構造表現の構成方法は2.1節参照）．

料理レシピに対する情報処理では，このような料理レシピの**木の全体**を用いることがしばしばである．たとえば類似レシピ検索や推薦ではレシピの木同士の何らかの意味での類似度や距離が用いられる [15, 18]．しかし実際には，料理レシピの**木の一部**に着目したい場合もある．例えば，用意できない食材や手元の機材では不可能な調理手順に注目し，その部分のみをその他の食材や方法で代替したい，というのはユーザの一般的なニーズであろう．あるいは，料理レシピを検索する際には，各レシピの特徴（他の類似レシピとの違いがよく表れた部分）が有用な情報になる．たとえばクックパッドで「肉じゃが」を検索すると 10,740 件[*5]のレシピがヒットするが，ユーザにとってその違いを見分けることは簡単でない．

前述した「レシピの木の一部に着目した情報処理」（代替食材・手法の提示，特徴的な部分の提示）をおこなうためには，どういった意味表現が必要だろうか？一点目として，レシピの木の中の**特徴的な部分木**（カタマリと考えて良い部分，置き換えて良い部分）を捉える必要があるだろう (図3)．レシピの木の中に意味的なまとまりは記載されていないため，レシピの木を教師なしで分割する必要がある．二点目として，そのように分割された**部分木同士の類似度をソフトに捉える**必要がある．代替食材・方法の提示においては，部分木同士をその類似度に基づいて置き換えることになると考えられるが，これをシンボルでおこなうのは困難である．たとえば「みじん切りにした長ネギ」と「細かく刻まれた葱」は文字列としては全く異なるがほとんど同じものを指している．本研究では，これらの二点のニーズを満たしたレシピの木の新しい意味表現を教師なしで学習することを試みる．

1.1 貢献

本研究の貢献は以下の通り：

- レシピの木を教師なしで分割し特徴的な部分木を捉える問題を，構文木の教師なし分割と捉えて確率木置換文法の学習によって実現した．
- レシピの部分木の連続表現を，分布仮説に基づく単語埋め込みや句の埋め込みと同様の手法で構成した．
- これらの手法を実際の大規模データに適用した際に学習される意味表現を確認した．

2 提案手法

本研究では，部分構造のまとまりを捉えながらそれらの連続的な類似度も考慮したレシピの意味表現を学習することを試みる．学習は 3 段階に分けておこなう：

1. 各レシピを表す木を構文木と考え，木置換文法の学習の枠組みでレシピの**特徴的な部分構造を取り出す**．
2. 各ノードに単語ベクトルと同様の分散表現を与え，その和によって**各部分構造に連続表現を付与する**．
3. 部分構造に付与された連続表現を用いてレシピの木の**中間ノードに対するシンボル細分化**を行う．

提案する学習手法の (1) 特徴的な部分構造の獲得および (2) 単語ベクトルの学習は，それぞれでこれらの巨大な生コーパスを活用できることを付記する．まず (1) の入力である「レシピを表す木」は，生テキストで書かれた大量のレシピを木構造に変換するパーザ [5] を活用することで自動生成することができる．エッジの種類を区別しない場合のエッジ推定の精度は 70% 台後半程度である[*6]．また (2) の単語ベクトルの学習でも，生

[*1] クックパッド（日本語）https://cookpad.com/，楽天レシピ（日本語）https://recipe.rakuten.co.jp/，GENIUS KITCHEN（英語）https://www.geniuskitchen.com/，愛料理（中国語）https://icook.tw/，など．
[*2] http://www.ieice.org/~cea/workshop/2018/
[*3] https://cookpad.com/
[*4] 中国語のレシピには，日本語や英語のレシピとは異なり，食材や調理方法と無関係のタイトルがつけられている場合がある．このため，レシピタイトルでの類似レシピ検索が困難であり，レシピの構造を活用する必要性が特に高い．
[*5] 2018 年 1 月 14 日現在
[*6] 著者に確認．

テキストで書かれた大量のレシピをそのまま活用できる．

2.1 データ

本研究では木構造レシピデータとして，172 万品の日本語レシピデータからなるクックパッドデータセット[*7]を一旦 Flow Graph 形式 [9] にパーズした上で，これを木構造に変換して用いる．手順は以下の通り．

1. **生テキスト → 情報付きテキスト**: 生テキストで書かれたレシピに対して，形態素解析器 KyTea[*8]を用いて形態素解析およびレシピ用語認識 [8] をおこない，各単語に形態素情報やレシピ用語の種別等の情報が付与された情報付きテキストにする．同時に，動詞や助動詞の終止形の復元用のファイルを作る．
2. **情報付きテキスト → フローグラフパーザ入力形式**: ステップ1で得られる情報付きテキストをフローグラフパーザに入力するため形式的に変換する．
3. **フローグラフパーザ入力形式 → フローグラフ**: フローグラフパーザ [5] を用いる．
4. **フローグラフ → レシピの木**:
   a) ステップ1で作成した終止形復元用ファイルを参照し，動詞や助動詞の終止形を復元する．
   b) 料理オントロジ [17] を用いて同義語を寄せる．例えば，「たまねぎ」「玉ねぎ」「玉葱」などの表現を「玉葱」という単一の表現に統一する．
   c) 食材 (F), 道具 (T), 調理者の動作 (Ac), 食材の動作 (Af) 以外のノードを削除する．
   d) 食材や道具ノードが連続している場合は，末端のノードのみを残す．たとえば，「(Ac(茹でる))→(F(芯))→(F(キャベツ))」は「(Ac(茹でる))→(F(キャベツ))」とする．
   e) 調理者の動作と共参照関係にある食材や道具ノードは削除する．たとえば，「(Ac(焼く))→(F(具)) $\underset{\text{equal}}{\rightarrow}$ (Ac(混ぜる))→ ...」は「(Ac(焼く))→(Ac(混ぜる))→ ...」とする．
   f) 各調理者の動作ノード (Ac) を中間生成物ノード (I) の子にする．たとえば，「(Ac(切る))(F(キャベツ)))」は「(I(Ac(切る))(F(キャベツ))」とする．

本稿における “レシピの木” のノードの意味

- F: 食材．子ノードとして食材の名前のノードを持つ．
- T: 道具．子ノードとして道具の名前のノードを持つ．
- Ac: 調理者の動作．子ノードとして調理者の動作の名前のノードを持つ．
- Af: 食材の動作．子ノードとして食材の動作の名前のノードを持つ．
- I: 中間生成物．子として 1 つの Ac または Af ノードと，0 個以上の F,T,I ノードを持つ．
- Root: 根ノード．子ノードの種類は I と同じ．

[*7] https://www.nii.ac.jp/dsc/idr/cookpad/cookpad.html
[*8] http://www.phontron.com/kytea/index-ja.html

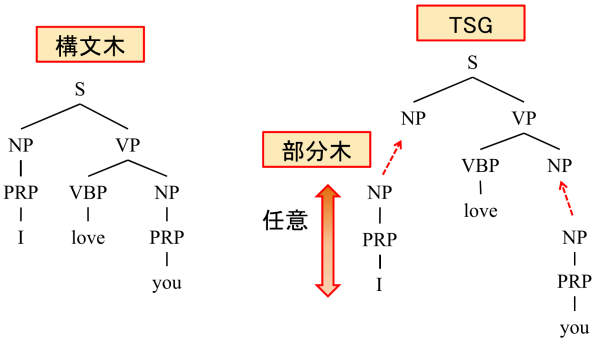

図1: 木置換文法 (図は進藤 (2012) https://www.hshindo.com/data/shindo-NLP2012-talk.pdf による．)

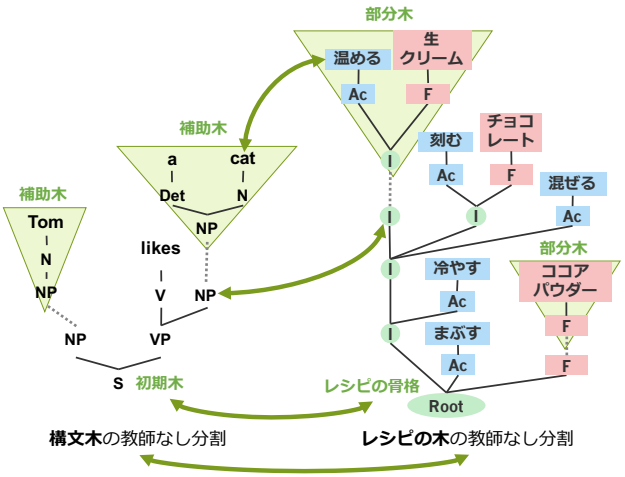

図2: 構文木の教師なし分割とレシピの木の教師なし分割の対応

2.2 木置換文法学習によるレシピの木の教師なし分割

レシピの特徴的な部分構造の獲得する問題（レシピの木の集合を入力として各木を教師なしで分割する問題）を，確率木置換文法の学習（構文木の集合を入力として各木を教師なしで分割する問題）と捉えて解く．

確率木置換文法（PTSG）とは，任意の大きさの部分木を基本単位とし，これらを組み合わせて構文木を構成する確率文法である1．

すなわち，構文木の生成確率 $p(\text{tree};\boldsymbol{\theta})$（構文木の導出の確率）が，構文木を構成する部分木の生成確率 $p(\text{subtree};\boldsymbol{\theta})$ の積（導出を構成する生成規則の確率の積）として与えられる：

$$p(\text{tree};\boldsymbol{\theta}) = \prod_{\text{subtree}\in\text{tree}} p(\text{subtree};\boldsymbol{\theta}). \quad (1)$$

各部分木の生成確率はこれをより単純化した部分木の生成確率によって補完（スムージング）することでゼロ頻度問題に対応できる．モデルパラメータ $\boldsymbol{\theta}$ の推定方法，および推定された $\boldsymbol{\theta}$ の下で尤もらしい木の分割（導出）を求める具体的な方法については，Cohn ら [2] や Shindo ら [12] を参照されたい．

レシピの木の特徴的な部分構造の学習（レシピの木の教師な

し分割）と確率木置換文法による文法抽出（構文木の教師なし分割）の対応は図 2の通り．すなわち，

- レシピの木 ↔ 構文木
- Root ノード ↔ 開始記号
- 食材 (F), 道具 (T), 調理者の動作 (Ac), 食材の動作 (Af), 中間生成物 (I) ノード ↔ 非終端記号
- レシピ用語ノード ↔ 終端記号

このように，レシピの木の集合を構文木の集合と見れば，レシピの特徴的な部分構造は文法抽出の枠組みで学習できることが分かる．本研究では木置換文法の学習に Julia による実装を用いた[*9]．

2.3 単語ベクトルの和による部分木の分散表現

2.2節で獲得したレシピの部分木間の類似性を測る際，部分木をシンボルとして扱うのは難しい．本節では，単語埋め込みや句の埋め込みと同様の枠組みで各部分木に分散表現を与える．

単語埋め込みと句の埋め込みについて簡単に触れる．生コーパス上における単語の共起情報を教師信号として学習された単語の分散表現（ベクトル表現）は今日の自然言語処理において不可欠な基本的ツールである [1, 7, 10, 11]．また，こうして学習された単語の分散表現の（単なる）和が，句や文の分散表現として有用に働くことが確認されている [6, 16]．

本研究では，単語埋め込みおよび句の埋め込みの学習手法に若干の工夫を加え部分木の分散表現を学習する．

レシピ用語の単語ベクトルの学習　事前準備として，生の自然言語文で書かれたレシピデータから，レシピ用語の単語ベクトルを学習する．アルゴリズムは word2vec，コーパスはクックパッドと国立情報学研究所が公開するデータセット[*10] を用いる．

終端記号　終端記号，たとえば“生クリーム”，“鍋”，“炒める”などの分散表現は，前述の単語ベクトルをそのまま用いる．

葉がすべて終端記号になっている部分木　葉がすべて終端記号になっている部分木，たとえば (F(生クリーム))，(T(鍋))，(Ac(炒める)),(I(Ac(茹でる))(I((Af(しんなりする))(F(キャベツ))))) などの分散表現は，終端記号（ここでは“茹でる”，“しんなりする”，“キャベツ”）の分散表現の相加平均を用いる．

葉に非終端記号が残っている部分木　葉に非終端記号が残っている部分木に関しては，

- **葉の終端記号**：終端記号の分散表現
- **葉の非終端記号（F，T，Ac，Af）**：訓練データ上でここから伸びている部分木（たとえば (F(人参))）たちの分散表現の平均
- **葉の非終端記号（I）**：無視[*11]

として，これらの平均を分散表現として与える．

[*9] https://github.com/hshindo/TSGs.jl

[*10] https://www.nii.ac.jp/dsc/idr/cookpad/cookpad.html
ただし，投稿されたレシピのデータを保持する recipes テーブルのみを使用した．

[*11] 相互再帰に陥る可能性があるため．

2.4 部分木のクラスタリング

フローグラフで用いられる中間ノードは F（食材）や T（道具）など比較的大きな単位でまとめられている．つまり，F というという共通の非終端記号（文脈）から，(F(鶏肉)) と (F(生クリーム)) という実用上全く別の文脈で用いられる部分木（食材）たちがともに生成されることになる．しかし実際には，例えばシュークリームのレシピの生成時は（お菓子を作るという文脈であれば），F という記号からは (F(生クリーム)) や (F(バター)) などお菓子に使い得る食材のみが生成されることが望ましい．本節では，前節で学習された部分木の連続表現に基づき，共通の非終端記号を根とする部分木たちを，肉類クラスタ $\{$(F(鶏肉)),(F(牛肉)),...$\}$ や乳製品クラスタ $\{$(F(生クリーム)),(F(バター)),...$\}$ にクラスタリングすることでこの問題に対処する．

具体的には，共通の非終端記号を根とする部分木たちに割り当てられた分散表現をそれぞれユークリッド距離で正規化した上で k-近傍法（ユークリッド距離）を適用する．実験ではクラスタ数 k をクラスタリング対象の部分木の数 n を用いて $\lceil \frac{n}{20} \rceil$ とした．

なお文法抽出においても同様の試みは見られ，非終端記号をサブカテゴリに細分化することで文脈情報を捉えた生成規則が学習できることが指摘されている [3, 12]．

2.5 部分木の葉へのクラスタ離散分布の割り当て

部分木のクラスタの情報を活用しながらレシピの木をサンプリングできるよう，各部分木の**葉**に割り当てられている中間ノード（F など）に，クラスタの離散分布を最尤推定で割り当てる．たとえば部分木

$$\text{(ROOT(AC(煮る))(I))} \tag{2}$$

の葉の I のクラスタ分布を推定する際は，まず訓練データ上で部分木(2)に接続されている I からはじまる部分木たちクラスタ番号を確認する．これがたとえば $\{\mathtt{I}_3, \mathtt{I}_{10}, \mathtt{I}_3, \mathtt{I}_3, \mathtt{I}_5, \mathtt{I}_3\}$ であれば，部分木(2)の葉 I にはクラスタ分布 $\{\mathtt{I}_3\colon \frac{4}{7}, \mathtt{I}_5\colon \frac{1}{7}, \mathtt{I}_{10}\colon \frac{1}{7}\}$ を割り当てる．

2.6 レシピの木の生成・サンプリング

レシピの木のサンプリング時は以下を再帰的に繰り返す．

1. 葉に割り当てられたクラスタの離散分布からクラスタ番号をサンプリング；
2. 当該クラスタの部分木たちから部分木をサンプリング（生成確率は当該の部分木たち全体で正規化）．

3 実験

提案手法により分割されたロールキャベツのレシピの例を図3に示す．ただし，例として挙げたレシピの部分木と同じクラスタに属する他の部分木については，紙面の都合上 2 つのみ掲載している．

木の分割に注目すると，細かい手順の集まりであるレシピの木をある程度の大きさをもつ“かたまり”に分割できていることがわかる．ただし，$\mathtt{I}_{22}, \mathtt{I}_{352}, \mathtt{I}_{477}$ のそれぞれを根とする木は「キャベツの下準備」としてまとめたほうが妥当であると考

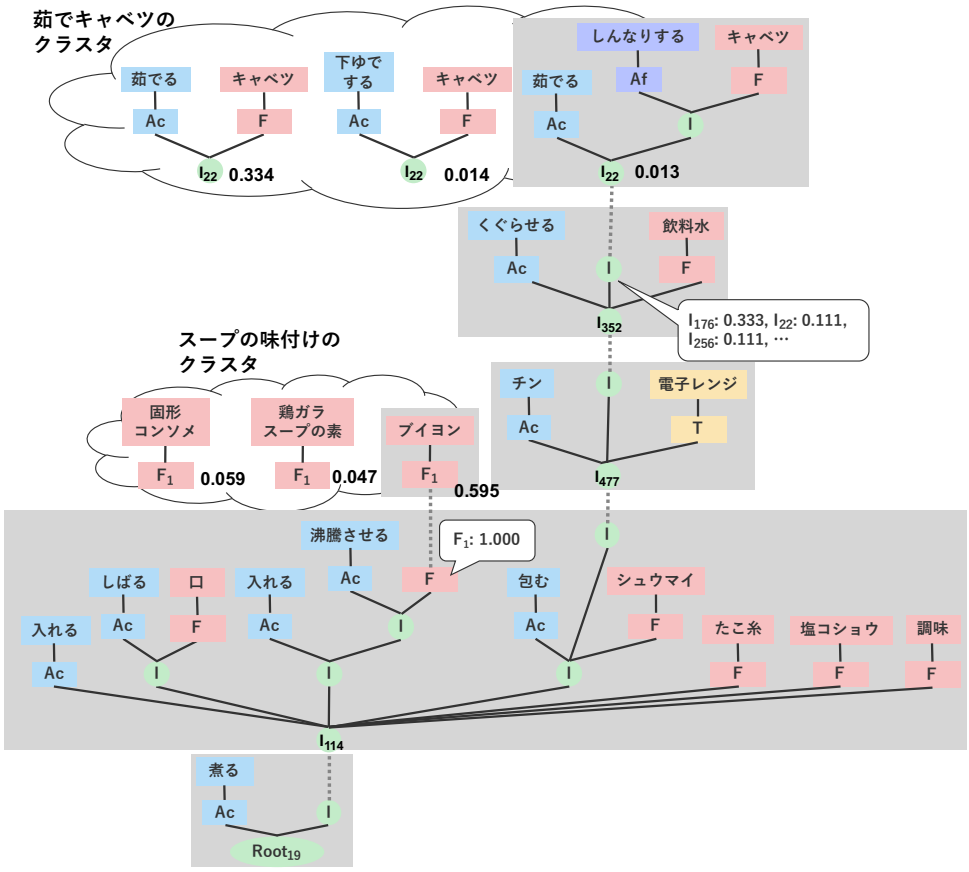

図3: 提案手法により分割された「ロールキャベツ」のレシピ

えられる．これに対しては，より高さのある部分木を選好する，階層的な分割を行う，などの学習アルゴリズムの改善が考えられる．

クラスタリングの結果に注目すると，スープの味付けをする食材や，キャベツを茹でる部分木が同じクラスタ (それぞれ ($\mathtt{F}_1, \mathtt{I}_{22}$) に入っていることがわかる．ただし，$\mathtt{I}_{22}$ には，春雨を茹でる部分木なども含まれている．これは，部分木に分散表現を割り当てる際に，葉の分散表現の平均をとっているためであると考えられる．つまり，提案手法では部分木の分散表現を計算する際に「その部分木には何が使われているか」は葉の分散表現を使うことで考慮しているが．「その部分木はどのように使われるか」を考慮していない．「どのように使われるか」の情報を取り入れることは今後の課題である．

謝辞　本研究の一部は JST CREST（JPMJCR1301）および株式会社日立製作所の支援を受けて行った．また本研究では，クックパッド株式会社と国立情報学研究所が提供する「クックパッドデータ」を利用した．

参考文献

[1] Piotr Bojanowski et al. “Enriching Word Vectors with Subword Information”. In: *Proc. of TACL* 5 (2017), pp. 135–146.
[2] Trevor Cohn, Phil Blunsom, and Sharon Goldwater. “Inducing tree-substitution grammars”. In: *Journal of Machine Learning Research* 11.Nov (2010), pp. 3053–3096.
[3] Michael Collins. “Head-driven statistical models for natural language parsing”. In: *Computational linguistics* 29.4 (2003), pp. 589–637.
[4] Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. “Globally coherent text generation with neural checklist models”. In: *Proc. of EMNLP*. 2016, pp. 329–339.
[5] Hirokuni Maeta, Tetsuro Sasada, and Shinsuke Mori. “A framework for procedural text understanding”. In: *Proceedings of IWPT*. 2015, pp. 50–60.
[6] Tomas Mikolov et al. “Distributed Representations of Words and Phrases and their Compositionality”. In: *Proc. of NIPS*. 2013, pp. 3111–3119.
[7] Tomas Mikolov et al. “Efficient Estimation of Word Representations in Vector Space”. In: *ICLR workshop*. 2013, pp. 1–12.
[8] Shinsuke Mori et al. “A machine learning approach to recipe text processing”. In: *Proc. of the 1st Cooking with Computer Workshop*. 2012, pp. 29–34.
[9] Shinsuke Mori et al. “Flow Graph Corpus from Recipe Texts.” In: *Proc. of LREC*. 2014, pp. 2370–2377.
[10] Jeffrey Pennington, Richard Socher, and Christopher D Manning. “GloVe: Global Vectors for Word Representation”. In: *Proc. of EMNLP*. 2014, pp. 1532–1543.
[11] Matthew E. Peters et al. “Deep contextualized word representations”. In: *Proc. of NAACL*. 2018, pp. 2227–2237.
[12] Hiroyuki Shindo et al. “Bayesian symbol-refined tree substitution grammars for syntactic parsing”. In: *Proceedings of ACL*. Association for Computational Linguistics. 2012, pp. 440–448.
[13] Han Su et al. “Automatic recipe cuisine classification by ingredients”. In: *Proc. of Ubicomp*. ACM. 2014, pp. 565–570.
[14] Liping Wang and Qing Li. “A personalized recipe database system with user-centered adaptation and tutoring support”. In: *ACM SIGMOD Ph. D. workshop on Innovative database research*. Citeseer. 2007.
[15] Liping Wang et al. “Substructure similarity measurement in chinese recipes”. In: *Proc. of WWW*. ACM. 2008, pp. 979–988.
[16] John Wieting et al. “Towards Universal Paraphrastic Sentence Embeddings”. In: *Proc. of ICLR*. 2016, pp. 1–19.
[17] 土居洋子 et al. “料理レシピと特許データベースからの料理オントロジーの構築”. In: 電子情報通信学会技術研究報告 113.470 (2014), pp. 37–42.
[18] 山肩洋子 et al. “ワークフロー表現を用いたレシピの典型性評価と典型的なレシピの生成”. In: 電子情報通信学会論文誌 *D* 99.4 (2016), pp. 378–391.