オープンドメインの手順書のフローグラフ予測とデータセットの構築

白井 圭佑 [1]　亀甲 博貴 [2]　森 信介 [2]
[1] 京都大学大学院情報学研究科　[2] 京都大学学術情報メディアセンター
shirai.keisuke.64x@st.kyoto-u.ac.jp　{kameko,forest}@i.kyoto-u.ac.jp

概要

機械による手順書の理解は, それらにおける推論や実世界における自動化を行う上で重要である. 先行研究では, 手順書の理解の表現としてレシピフローグラフという枠組みが提案されている. 本稿では, レシピにおける食材を手順書が目指す最終成果物の構成要素として捉えることで, 調理以外の分野への適用を試みる. また, 調理以外の分野におけるフローグラフの予測精度を調査するため, 新たにwikiHow Flow Graph (wikiHow-FG) コーパスを構築した. コーパスは, wikiHow から選択した 2 分野を対象に, 30 記事ずつアノテーションすることで構築した. 実験では, 対象分野の少量データでモデルの学習を行う few-shot の設定と, 既存の調理分野のデータで学習を行う zero-shot の設定を考え, それぞれについて評価, 比較を行った.

1　はじめに

手順書は調理や家具の組み立て等, 日常生活における作業を補助する目的で書かれた文書である. 機械による手順書の理解は, 手順における推論 [1] やその作業の自動化 [2] を目指す上で重要であるといえる. これには, 文書全体における重要表現の認識やそれらの係り受け関係を解析する必要がある. 手順書のグラフ表現への変換 [3] は, これらの問題を解決する一つの手段である.

先行研究では, この方向において分野依存の手法が提案されてきた [4, 5, 6]. 特に, 調理分野においてはウェブ上に豊富な資源が存在することから, 様々なアプローチが取られてきた [4, 7, 8]. この中で, 森ら [4] はレシピをフローグラフへ変換する枠組みとして, レシピフローグラフと日本語コーパスの提案を行った. 後の研究 [9] では, 英語フローグラフ (English Flow Graph; English-FG) とそのコーパスも提案されている. 一方でレシピフローグラフはアノテーションコストが高く, 大規模なデータセットの構築は現実的ではないという問題がある. また, その適用分野も現状は調理のみに留まっている.

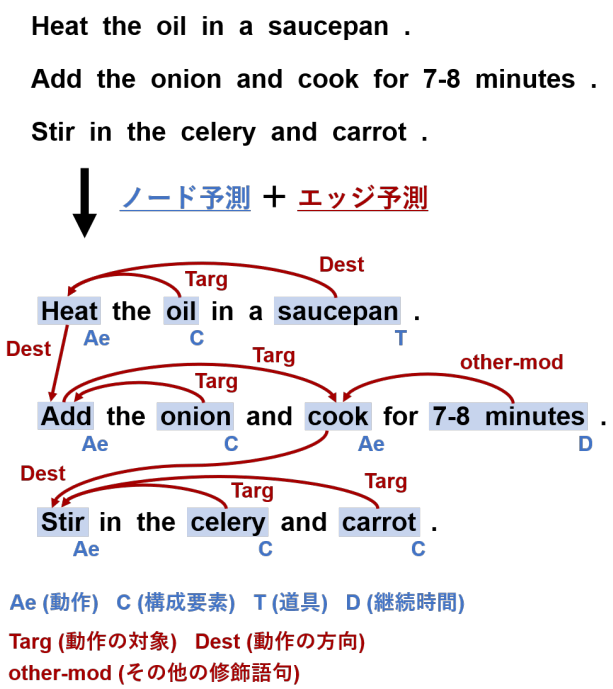

図 1　フローグラフ予測の例. タグとラベルは提案フレームワークのものである. 本研究では, このフレームワークを用いて調理以外の分野における手順書のフローグラフ予測を行う.

本研究では, レシピにおける食材を手順書が目指す成果物の構成要素として捉えることで, 調理以外の分野への適用を目指す. 本稿では, 英語の手順書を対象とし, 既存研究 [10] と同様にノード予測とエッジ予測の 2 段階からグラフの予測を行う. また, 調理以外の分野におけるフローグラフの予測精度を調査するため, 新たに wikiHow-FG コーパスを提案する. これは, 様々なタスクに関するマニュアルを公開している wikiHow から記事を収集して構築した. 対象分野として, 表 1 に示す 2 分野を wikiHow

表 1　各記事における記事タイトルの例.

分野	記事タイトルの例
Hobbies and Crafts	*Making a bar soap*, *Making a duct tape bow*, *Making a paper box*
Home and Garden	*Cleaning a mattress pad*, *Installing a microwave*, *Making a scented candle*

表 2　タグとその意味. 括弧内は English-FG にのける名称と定義を指す.

タグ	意味
C (F)	構成要素 (食材)
T	道具
D	継続時間
Q	分量
Ae (Ac)	動作 (調理動作)
Ae2 (Ac2)	不連続の動作 (調理動作)
Ac (Af)	構成要素 (食材) による動作
At	道具による動作
Sc (Sf)	構成要素 (食材) の状態
St	道具の状態

のカテゴリから選択し, 各分野において 30 記事をアノテーションした. 実験では, 既存の English-FG コーパス [9] 上で学習を行う zero-shot の設定と, wikiHow-FG 上の少量のアノテーションで学習を行う few-shot の設定における予測精度を, それぞれ調査した. 結果から, ノード予測に関しては zero-shot, few-shot 共に高い予測精度が得られる一方で, エッジ予測に関しては zero-shot が few-shot モデルの予測精度を大幅に上回る結果となった.

2　フローグラフ表現

2.1　レシピフローグラフ

ここでは, レシピフローグラフについて概説する. レシピフローグラフは, 図 1 に示すような有向非巡回グラフである. ここで, 点は文書中の重要表現を指し, 表 2.1 に示すような 10 種類のタグを用いて表す. 辺は点同士の関係を指し, 表 3 に示すような 13 種類のラベルを用いて表す. レシピフローグラフの予測手法として, 先行研究 [10] ではノード予測とエッジ予測の 2 段階に分けて行うことが提案されている.

ノード予測では文章中の重要表現とそれに対応するタグの識別を行う. これは系列ラベリング問題として定式化され, 固有表現認識器 (Named Entity Recognition; NER) を用いてタグの予測を行う. また, NER は文レベルでのタグ予測が一般的である [11] が, 先行研究 [9] では文章全体を入力としており, 本研究でもそれに従う[1].

表 3　ラベルとその意味. 括弧内は English-FG における名称と定義を指す.

ラベル	意味
Agent	主語
Targ	動作の対象
Dest	動作の方向
T-comp	道具による補足
C-comp (F-comp)	構成要素 (食材) による補足
C-eq (F-eq)	同一の構成要素 (食材)
C-part-of (F-part-of)	構成要素 (食材) の一部
C-set (F-set)	構成要素 (食材) の集合
T-eq	同一の道具
T-part-of	道具の一部
A-eq	同一の動作
V-tm	動作のタイミング
other-mod	その他の修飾語句

エッジ予測ではノード間における依存関係を依存関係ラベルと共に予測することでグラフの構築を行う. これは最大全域木問題として以下のように定式化される.

$$\hat{G} = \underset{G \in \mathcal{G}}{\operatorname{argmax}} \sum_{(u,v,l)} s(u, v, l). \tag{1}$$

ここで, $s(u, v, l)$ は点 u から点 v への辺がラベル l を持つ場合のスコアを表す. この問題は Chu-liu edmonds アルゴリズムを用いて解くことが出来, スコアは依存構造解析器 [12] を用いて計算する.

2.2　調理以外の分野への適用

既存の English-FG [9] を基に, レシピにおける食材を手順書の成果物における構成要素として捉えることで, 他分野への適用を行う. ここでの構成要素とは, 例えば調理ではトマトや牛肉等, クラフトでは段ボールや糊等が対応し, レシピフローグラフにおける定義と同様に道具とは区別して扱う. 本研究で用いるタグと依存関係ラベルの定義を表 2.1 と表 3 にそれぞれ示す. これらは English-FG の定義を一部

1)　English-FG コーパスにおける予備実験では, 文章レベルで予測を行うことで, 文レベルで予測を行った場合に比べ約 10%の精度向上が見られることを確認した.

表 4　コーパスの統計量.

分野	文字数	単語数	手順数	タグ数	ラベル数
Hobbies and Crafts	9,407	2,556	247	1,062	1,076
Home and Garden	7,700	2,010	205	894	886

修正したものであり, タグやラベルの追加や削除は行っていない. また, 構成要素を食材に置き換えることで, 既存の調理分野に適用することも可能である. また, フローグラフの予測に関しては, 先行研究 [10] と同様に, ノード予測とエッジ予測の 2 段階に分けて行う.

3　wikiHow-FG コーパス

wikiHow-FG コーパスは wikiHow[2] から収集した記事に対し, フローグラフのアノテーションを行うことで構築した. wikiHow は 11 万を超えるマニュアルを公開しているウェブサイトであり, 手順に関する言語資源として近年注目を集めている [13, 1, 14, 15]. 以下では, データ収集方法, アノテーション手順, コーパスの統計量について順に説明を行う.

3.1　データ収集

本研究では, 表 1 に示す wikiHow の 2 カテゴリを対象分野として選択した. ここで, *Hobbies and Crafts* は材料の組み立てタスクを主に対象としており, 食材の組み立てを行う調理分野とは比較的近いといえる. 一方で, *Home and Garden* では修理や掃除等, 組み立てではないタスクを主に対象とするため, より調理から離れた分野であるといえる. 記事の収集は, これらの各分野に含まれる 30 記事を, 既存の wikiHow Corpus [1] から収集する形で行った. また, 低品質な記事を省くため, 全体で 25 単語以下のもの, ユーザー評価が 50%以下のものは収集の対象外とした. また, 抽象的な目的を目指す記事や, 曖昧な表現の多い記事を手動で除去した. 以降では, 収集した記事に含まれる見出し (headline) を手順として使用する.

3.2　アノテーション

先行研究で提案されたアノテーションツール [16] を用いて, アノテーションを行った[3]. また, English-FG コーパスから収集した 10 記事を用いて, 事前にアノテータのトレーニングを行った. コーパスの統計量を表 4 に示す. 表から, 収集した記事は 1 記事辺り平均 7.53 手順 (76.1 単語) で構成されていることがわかる. また, 1 記事あたりの平均アノテーション数が, 32.6 タグと 32.7 ラベルであることも見て取れる.

4　実験

構築した wikiHow-FG コーパスと既存の English-FG コーパスを用いて, ノード予測とエッジ予測の実験をそれぞれ行った. 本研究では, English-FG のみで学習を行い, 対象分野で予測を行う zero-shot の設定と, 対象分野の少量の wikiHow-FG データで学習を行い, 予測を行う zero-shot の設定を想定し, それぞれにおける予測性能を調査した. フローグラフはアノテーションコストが高いため, これらは現実的な実験設定であるといえる.

4.1　ノード予測

4.1.1　実験設定

NER モデルとして BiLSTM-CRF [11] を用いた. また, 元のモデルにおける BiLSTM エンコーダを事前学習済みの DeBERTa [18] に差し替えて用いた[4]. また学習時には DeBERTa のパラメータも調節の対象とした.

最適化には AdamW [20] を, 初期学習率を 1.0×10^{-5}, weight decay の係数を 1.0×10^{-5} として用いた. また, 各イテレーションにおいて, Cosine Annealing [20] (S_d 回) と Linear Warmup (S_w 回) を組み合わせて学習率の調節を行った. ミニバッチサイズは B 記事から構築し, English-FG 上での学習の際には $(B, S_\mathrm{w}, S_\mathrm{d}) = (5, 500, 4500)$ に, wikiHow 上での学習の際には $(B, S_\mathrm{w}, S_\mathrm{d}) = (3, 100, 900)$ に, 設定した. これらのハイパーパラメータの調節は開発セット上で行った.

wikiHow-FG 上の学習には, 全体 30 記事を 6 分割し, 1 分割を学習セット, 1 分割を開発セット, 残り 4 分割をテストセットとすることで, 6-分割交差検証を行った. English-FG 上の学習には全体 300 記事

2)　https://www.wikihow.com
3)　事前に stanza [17] を用いて手順に含まれる文の単語分割を行った.
4)　先行研究 [9] では BERT [19] が用いられている. English-FG コーパスにおける予備実験では, DeBERTa を用いることで 0.47%の精度向上が実現出来ることを確認した.

表 5 ノード予測における実験結果. チェックマーク (✓) は使用した学習データを指す.

対象分野	学習データ		精度	再現率	F 値
	English -FG	wikiHow -FG			
Hobbies and Crafts		✓	0.698	**0.707**	**0.702**
	✓		**0.703**	0.684	0.693
Home and Garden		✓	0.663	0.676	0.669
	✓		**0.734**	**0.742**	**0.738**

表 6 エッジ予測の実験結果. チェックマークは使用した学習データを指す.

対象分野	学習データ		精度	再現率	F 値
	English -FG	wikiHow -FG			
Hobbies and Crafts		✓	0.285	0.281	0.283
	✓		**0.613**	**0.605**	**0.609**
Home and Garden		✓	0.229	0.232	0.231
	✓		**0.644**	**0.649**	**0.646**

のうち, 80%を学習セット, 10%を開発セット, 残り 10%をテストセットとすることで 10-分割交差検証を行った. English-FG 上で学習したモデルの評価の際には, wikiHow-FG の各分割におけるテストセットに対し, それぞれ 1 モデルを割り当てて予測を行った. 実験結果では, 対象分野の wikiHow-FG コーパスの各テストセットにおけるスコアの平均を報告する. 評価指標としては, 精度, 再現率, F 値を用いた.

4.1.2 実験結果

表 5 に結果を示す. 両分野において, zero-shot, few-shot の設定ともに 66%以上の精度, 再現率, F 値が実現出来ていることがわかる. 特に few-shot 設定における結果は, 対象分野における数記事のアノテーションを用意することで, 比較的高精度なノード予測モデルが実現可能であることを示唆している. また, *Home and Garden* においては, zero-shot 設定のモデルが few-shot 設定のモデルを, 全指標において 6.8%以上も上回る結果となった. これに関しては, 調理分野と *Home and Garden* が, *Hobbies and Crafts* よりも近い分野であり, 表現やタグの予測が *Hobbies and Crafts* 以上に容易だったのではないかと思われる.

4.2 エッジ予測

4.2.1 実験設定

依存構造解析器として Biaffine Attention Parser [12] を使用した[5]. 単語の分散表現は, 事前学習済み DeBERTa を用いて計算した. ノード予測の時と同様に, DeBERTa のパラメータも調節の対象とした.

最適化は AdamW [20] を用い, 各イテレーション毎に Cosline Annealing と Linear Wamup を用いた学習率の調節を行った. ハイパーパラメータに関しては, 4.1.1 節と同様の値を用いた. 学習は English-FG コーパス, wikiHow-FG コーパス共に 4.1.1 節と同様の分割を用い, それぞれ交差検証を行った. 評価指標には, 予測結果と正解データ間におけるラベル付き辺の精度, 再現率, F 値を用いた.

5) 先行研究 [9] では, 線形モデル [10] による解析器を使用している. English-FG コーパスを用いた予備実験では, 本モデルを用いることで 2.7%の精度向上が実現できることを確認した.

4.2.2 実験結果

表 6 に結果を示す. ノード予測の実験結果 (4.1.2 節) と異なり, zero-shot 設定のモデルの性能が著しく低い結果となっていることがわかる. また, few-shot 設定のモデル間とのスコアの差も大きく, F 値に関して 32.6%以上の開きがある. これは, エッジ予測モデルの学習には, ノード予測モデル以上に多くの例が必要であることを意味しており, 結果として学習データがより豊富な zero-shot 設定の方が高いスコアが得られたのではないかと考えられる.

5 おわりに

本稿では, 既存のレシピフローグラフを利用し, 他分野の手順書のフローグラフ予測を試みた. また, そのために wikiHow の 2 分野を対象に収集した記事を基に, 新たに wikiHow-FG コーパスを構築した. 実験では, 既存の調理分野のデータとして English-FG コーパスのみで学習を行う zero-shot 設定と wikiHow-FG コーパスの対象分野のデータのみを用いた few-shot 設定とで, 個別にモデルを学習し, 評価を行った. 結果からは, ノード予測においては両設定ともに 66%以上のスコアが得られる一方で, エッジ予測においては, zero-shot 設定のモデルの方が高い予測性能を実現出来ることがわかった.

今後の展望としては, 本稿で調査した以外の分野でも調査を行う他, 調理分野から対象分野への分野適応を行うことで性能向上を狙うことが挙げられる. また, 対象分野のデータ拡張を行うことで性能を向上させることも考えられる.

参考文献

[1] Li Zhang, Qing Lyu, and Chris Callison-Burch. Reasoning about goals, steps, and temporal ordering with WikiHow. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing**, pp. 4630–4639. Association for Computational Linguistics, 2020.

[2] Mario Bollini, Stefanie Tellex, Tyler Thompson, Nicholas Roy, and Daniela Rus. Interpreting and executing recipes with a cooking robot. In **Experimental Robotics**, pp. 481–495. Springer, 2013.

[3] Yoshio Momouchi. Control structures for actions in procedural texts and PT-chart. In **COLING 1980 Volume 1: The 8th International Conference on Computational Linguistics**, 1980.

[4] Shinsuke Mori, Hirokuni Maeta, Yoko Yamakata, and Tetsuro Sasada. Flow graph corpus from recipe texts. In **Proceedings of the Ninth International Conference on Language Resources and Evaluation**, pp. 2370–2377, 2014.

[5] Chaitanya Kulkarni, Wei Xu, Alan Ritter, and Raghu Machiraju. An annotated corpus for machine reading of instructions in wet lab protocols. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)**, pp. 97–106. Association for Computational Linguistics, 2018.

[6] Fusataka Kuniyoshi, Kohei Makino, Jun Ozawa, and Makoto Miwa. Annotating and extracting synthesis process of all-solid-state batteries from scientific literature. In **Proceedings of the Twelfth Language Resources and Evaluation Conference**, pp. 1941–1950. European Language Resources Association, 2020.

[7] Liang-Ming Pan, Jingjing Chen, Jianlong Wu, Shaoteng Liu, Chong-Wah Ngo, Min-Yen Kan, Yugang Jiang, and Tat-Seng Chua. **Multi-Modal Cooking Workflow Construction for Food Recipes**, p. 1132–1141. Association for Computing Machinery, 2020.

[8] Dim P. Papadopoulos, Enrique Mora, Nadiia Chepurko, Kuan Wei Huang, Ferda Ofli, and Antonio Torralba. Learning program representations for food images and cooking recipes. In **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**, pp. 16559–16569, 2022.

[9] Yoko Yamakata, Shinsuke Mori, and John A Carroll. English recipe flow graph corpus. In **Proceedings of the 12th Language Resources and Evaluation Conference**, pp. 5187–5194, 2020.

[10] Hirokuni Maeta, Tetsuro Sasada, and Shinsuke Mori. A framework for procedural text understanding. In **Proceedings of the 14th International Conference on Parsing Technologies**, pp. 50–60, 2015.

[11] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In **Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 260–270. Association for Computational Linguistics, 2016.

[12] Timothy Dozat and Christopher D. Manning. Simpler but more accurate semantic dependency parsing. In **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)**, pp. 484–490. Association for Computational Linguistics, 2018.

[13] Yilun Zhou, Julie Shah, and Steven Schockaert. Learning household task knowledge from WikiHow descriptions. In **Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)**, pp. 50–56. Association for Computational Linguistics, 2019.

[14] Li Zhang, Qing Lyu, and Chris Callison-Burch. Intent detection with WikiHow. In **Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing**, pp. 328–333, 2020.

[15] Shuyan Zhou, Li Zhang, Yue Yang, Qing Lyu, Pengcheng Yin, Chris Callison-Burch, and Graham Neubig. Show me more details: Discovering hierarchies of procedures from semi-structured web data. In **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 2998–3012. Association for Computational Linguistics, 2022.

[16] Keisuke Shirai, Atsushi Hashimoto, Taichi Nishimura, Hirotaka Kameko, Shuhei Kurita, Yoshitaka Ushiku, and Shinsuke Mori. Visual recipe flow: A dataset for learning visual state changes of objects with recipe flows. In **Proceedings of the 29th International Conference on Computational Linguistics**, pp. 3570–3577. International Committee on Computational Linguistics, 2022.

[17] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A python natural language processing toolkit for many human languages. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations**, pp. 101–108. Association for Computational Linguistics, 2020.

[18] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: decoding-enhanced bert with disentangled attention. In **9th International Conference on Learning Representations**, 2021.

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long and Short Papers)**, pp. 4171–4186. Association for Computational Linguistics, 2019.

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In **Proceedings of the 7th International Conference on Learning Representations**, 2019.