

# 検索・分析のための手順文章からの意味構造抽出

前田 浩邦<sup>†</sup> 山肩 洋子<sup>†</sup> 森 信介<sup>††</sup>

<sup>†</sup> 京都大学情報学研究科 〒 606-8501 京都市左京区吉田本町

<sup>††</sup> 京都大学学術情報メディアセンター 〒 606-8501 京都市左京区吉田本町

E-mail: <sup>††</sup>forest@i.kyoto-u.ac.jp

あらまし より賢い検索や分析を可能とするために、手順文章から意味構造を抽出する手法を提案する。具体的には、手順文章としてレシピを題材とし、手順を記述している文章に対して、単語分割、固有表現認識、文章解析を順に行う。単語分割と固有表現認識は、既存手法を用いることとし、本論文では、意味構造の表現としてフローグラフ（無閉路有向グラフ）を提案し、単語に分割されレシピ固有表現タグが付与された文章をフローグラフに変換する文章解析について述べる。

キーワード 手順文章, レシピ, 意味構造, 自然言語処理

## 1. はじめに

テキストを対象とする検索や分析においては、自然言語処理を用いてその精度や利便性を高めることが一般的である。しかしながら、実用になっている自然言語処理は、単語分割や読み推定など、系列予測に分類される処理がほとんどである。ウェブページなどのテキストの構造や内容を計算機が理解できればより知的な検索や分析が可能となると考えられる。本論文では、対象を手順書（手続きを記述した文章）に限定し、自動的に文章全体の意味内容をフローグラフ表現に変換する手法を提案する。

従来の自然言語処理の研究は、様々な部分問題を設定し、各々を逐次的に組み合わせることにより言語処理を構築していた。文章理解を目的として、形態素解析（単語分割と品詞推定）、係り受け解析、述語項構造解析、共参照解析を逐次的に組み合わせる。本論文では、単語分割および重要語認識の結果を入力とし、重要語間の関係を表すフローグラフを出力する方法を提案する。

文章一般に対して内容表現の形式の良い定義を与えるのは非常に難しく、本論文では検索や分析の需要があり利便性が高いと考えられる手続き文書に限定する。手順書とは、物を作ったり、決められた状態にするまでの一連の手続きを自然言語で記述したものである。典型的な例は、取扱説明書や料理のレシピである。手順書は、複数のステップからなり、各ステップは複数の文からなる。各文は、どの物体をどの動作に用いるかということや、ある動作で得られる結果が後段のどの動作に使われるのかということなどを記述している。手順書を、その手続きの流れを示すフローグラフで表現することにより、物体と動作の間や、動作と動作の間との関係が明確になる。

本研究では、具体的に処理する手順書として料理のレシピ（以下単にレシピと呼ぶ）を採用する。レシピは、自然言語処理の観点からは、様々な言語現象を含んでおり興味深い解析対象である。応用の観点からは、多数のインターネットサイトがあり、加えて書店でも大きなスペースが割り当てられているなど、

レシピはもっとも需要がある手順書と考えられる。先行研究として、レシピテキストのフローグラフ化はより詳細なレシピテキストの検索に有効であることが示されている [1]。さらに、レシピテキストとその実施映像をそれぞれ解析して 2 つのフローグラフを生成し、両者をマッチングすることによって、言語表現と映像領域の対を自動で獲得する研究が行われている [2]。

本論文では、手順書をフローグラフに自動変換するための機械学習に基づく手法を提案する。実験にはレシピとそのフローグラフ表現からなるコーパス [3] [4] を用いる。フローグラフの頂点は、手順書中の重要語である。辺は重要語間の関係を表し、従来の自然言語処理においては係り受け関係、述語項構造、共参照・照応に相当する関係である。初期のレシピの構造化の研究 [5] では、フローグラフとして木が用いられている。しかし、材料の分割などのように、手順書の手続きを表現するには、無閉路有向グラフ (DAG; Directed Acyclic Graph) が必要である。上述のレシピフローグラフコーパスでは、内容表現として最終生成物を根とする根つきの DAG が用いられている。

フローグラフへ変換手法は、係り受け解析において使われてきた手法 [6] [7] の拡張である。本手法は、手順書内の重要語があらかじめ抽出されていると仮定する。その状態で、重要語の間に辺を張り、フローグラフを出力する。本研究で扱う言語現象は、述語項構造、係り受け、共参照などを含んでいる。従来は、各々を解析する処理を逐次的に組み合わせて処理を構築していたが、本手法では、一度に処理を行い、フローグラフを得る。機械学習に基づく提案手法の有効性を検証するために、ルールに基づいたフローグラフの推定手法と比較する。さらに、上述した言語現象の観点から本手法による推定結果を分析する。

## 2. 関連研究

図 1 に示すように、レシピテキストのフローグラフ化には単語分割と固有表現認識が必要である。Mori ら [8] は、レシピテキストをフローグラフに変換することを目的とし、変換のために必要な言語処理を調理ドメインに適応させている。Mori らは、点予測と部分的にアノテーションされたコーパスを使うこ

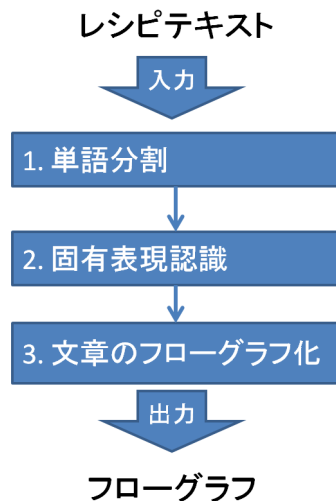


図 1 レシピテキストのフローグラフへの変換の流れ

とで、調理ドメインでの単語分割、固有表現認識、係り受け解析の精度をそれぞれ上げている。ただし Mori らは個々の解析器の精度の向上を示しただけであり、レシピテキスト全体のフロー化には至っていない。

レシピテキストからフローグラフへの変換手法はいくつか提案されている。Momouchi [9] は、人間が定義したバックスナウア記法に基づいた、レシピなどを含む手続き文章全般の解析手法を提案している。Hamada ら [10] は、レシピテキストから木に自動変換する手法を提案している。この手法では、まず人手で素材、調味料、動作などの属性の入った辞書を構築しておく。次に、レシピテキストに対して辞書との最大マッチングを行いノードを取り出す。最後に、述語項構造に従ってノード間に辺を張り、レシピテキストをグラフに変換する。Hamada らの研究は先駆的ではあるが、この手法は整形されたテキストに適用されることを前提としており、様々なスタイルの文章や表現に対して頑健ではないという問題点がある。山肩ら [2] は、レシピテキストを木に変換し、レシピテキストと調理映像の対応付けをする手法を提案する。この対応付けにより、レシピテキストから調理映像を参照できるような調理コンテンツの構築が可能になる。第 1 節でも述べたように、山肩らの木の生成手法は規則に基づくものであり、頑健な解析は難しい。

我々は係り受け解析の手法を使ってレシピ全体をグラフ化する。係り受け解析には大きく分けて 2 種類のアプローチがある。すなわち、グラフベースのアプローチ [11] [6] と遷移ベースのアプローチ [7] [12] である。グラフベースのアプローチの代表的な手法は最大全域木 (MST; Maximum Spanning Tree) を用いるものである。MST は McDonald ら [6] によって文の係り受け解析へ導入された。現在では係り受け解析の主要な手法の一つとして知られており [13]、MST をベースにした応用も研究されている [14]。McDonald ら [13] は、これらグラフベースと遷移ベースの 2 つのアプローチを比較し、全体的な精度はどちらの手法も大差ないものの、様々な係り受けの場合の中でそれぞれ精度が上がる場合と下がる場合があることを示した。例えば、McDonald らによると、遷移ベースのアプローチは、

表 1 固有表現の種類

タグ	意味	備考
F	食材	中間・最終生成物を含む
T	道具	調理道具や器など
D	継続時間	概数表現を含む
Q	分量	概数表現を含む
Ac	調理者の動作	語幹のみ
Af	食材の動作	語幹のみ
Sf	食材の状態	
St	道具の状態	

誤差伝播が起きることにより、長い文に対する係り受け精度がグラフベースのアプローチよりも低くなる傾向がある。

### 3. レシピフローグラフ

本節では、レシピフローグラフコーパス [3] [4] におけるフローグラフの定義を概説する。

#### 3.1 レシピとフローグラフ

通常、レシピは以下の 3 つの部分から構成される。

- (1) タイトル
- (2) 食材リスト (材料名および分量)
- (3) 調理手順 (手順番号および手順説明文)

本稿で説明するフローグラフは、調理手順の部分の意味内容を表現する。レシピの例を図 2 に、この調理手順に対応するフローグラフを図 3 にそれぞれ示す。

レシピで記述される手順には、依存関係があるものもあれば、ないものもある。例えば、調理者は、ニンジン茹でる前にニンジン切る必要があるが、ニンジン茹でる前にジャガイモ切る必要はない。そこで、調理の手順を表現するために、フローグラフを用いる。ほとんどのレシピは木で表現できるが、木で表現できないレシピも存在する。典型的な例は、食材の分割である。したがって、フローグラフは、無閉路有向グラフ (DAG; Directed Acyclic Graph) である。

フローグラフの頂点は、食材や道具、あるいは動作などに対応する。辺は、それらの関係を表す。一般の DAG とは異なり、レシピフローグラフは根と呼ばれる特別な頂点がある。これは、料理の完成品に対応する。以下、頂点と辺についての説明を行う。

#### 3.2 フローグラフの頂点

フローグラフの頂点は、食材や動作を表す単語列からなる。通常、ある実態に対応する単語列を固有表現と呼ぶので、本コーパスにおいてもこれを踏襲する。表 1 に、固有表現の種類 (クラス) を掲げる。この表からわかるように、通常の固有表現と異なり、調理者や食材の動作を表す用言も含む。以下では、各固有表現クラスについて順に説明する。例の表記にあたっては、単語境界を空白で示し、固有表現となる単語列をスラッシュ (//) で囲み、最後にクラスを示す記号を付加している。

##### 3.2.1 F: 食材

食材は、材料リストに記載されているような材料名に加えて、「タネ」「生地」といった中間食材名や、料理の完成品を含

料理名: なんなんなん♪

**材料(二人分)**

- ☆トマト約2個
- ☆たまねぎ(小)約1個
- ☆七味唐辛子適宜
- ☆オリーブオイル適宜
- ☆バジル適宜
- ☆酢適宜
- ☆レモン汁適宜

☆イタリアン  
エッセンス適宜  
☆豆板醤適宜  
☆塩適宜  
キャベツ少量  
ピーマン約1個  
とろけるチーズ4枚  
ウィンナー4本  
ケチャップ適宜

**作り方**

- 1 サルサソースを作る。トマトは角切り、たまねぎはみじん切りにし、☆の材料を全てボールにいれて混ぜる。味を見ながら調味料を加えてください。
- 2 キャベツは千切り、ピーマンは輪切りにする。
- 3 ナンにとろけるチーズを巻いたウィンナーをのせ、かる〜くオーブントースターでチンする。
- 4 チーズのとけた上から、サルサソース、切った野菜(キャベツ、ピーマン)をのせる。
- 5 ケチャップを好きなようにかけてめしあがれ♪

図2 レシピの例

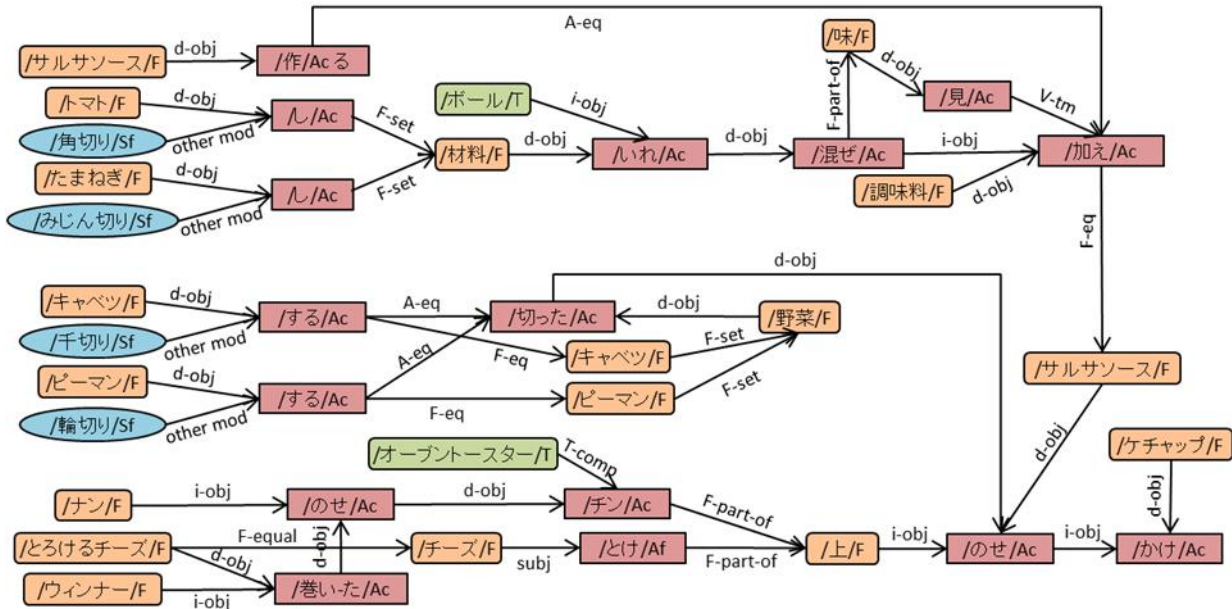


図3 図2のレシピから生成されたフローグラフ

表2 辺のラベル

ラベル	意味
subj	主語 (ガ格)
d-obj	対象 (ヲ格)
i-obj	方向 (ニ格)
T-comp	道具デ
F-comp	食材デ
F-eq	同一の食材
F-part-of	食材の一部
F-set	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	動作を行う
other-mod	その他の修飾語句

む。ただし修飾句はFに含めない。典型的な例は、以下の通りである。

例) 新しい /卵/F を /割/Ac っ て

3.2.2 T: 道具

鍋、蓋、包丁、コンロなど、調理道具や器を道具とする。手や指などの調理者の身体の一部も道具のような振る舞いをする

場合があり、Tとする。

例) /3分/D /レンジ/T を /し/Ac て から

3.2.3 D: 継続時間

加熱時間や冷却時間など、加工の継続時間を示す。数字と単位のほか、それらに対する概数表現等も含める。

例) /12 ~ 15分/D 間 /煮込/Ac み ます

例) /5分くらい/D

3.2.4 Q: 分量

食材の量を指定する表現を分量とする。数字部分や単位に加えて、それらに対する概数表現等も含める。

例) /人参/F /3 ~ 4cm くらい/Q

3.2.5 Ac: 調理者の動作

調理者が主語となる用言で、主に他動詞である。表層表現の比較により同一性を判定できるようにするために、活用語尾を含めない。また、補助的動詞は含めない。

例) /切/Af っ た /ニンジン/F を

例) /沸騰 させ/Ac たら

### 3.2.6 Af: 食材の動作

食材が主語となる用言もある。その場合には、Afとする。範囲や判定の基準はAcと同じである。

### 3.2.7 Sf: 食材の様態

用意された食材様態の初期状態を表す表現、並びにAcやAfの影響で遷移する(した)状態を表す表現をSfとする。ただし、量の表現はQである。

例) /柔らか/Sf < /な/Af する

例) /色/Sf が /変わ/Af する

例) /濃いめ/Sf の /調味料/F

### 3.2.8 St: 道具の様態

用意された道具様態の初期状態を表す表現、並びにAcやAfの影響で遷移する(した)状態を表す表現をStとする。

例) /強火/St の /フライパン/T で /炒め/Ac

例) /オープン/T を /150度/St に /予熱/Ac

## 3.3 レシピフローグラフの辺

前節で述べた固有表現について、文章中の固有表現間に以下であげる3つのうちのいずれかの関係が認められる場合、それらの固有表現に対応する頂点の間に枝を張り、ラベルとしてその関係の種類を付与する。

- 述語項関係がある場合(ゼロ照応も含む)
- 同種の2つの固有表現の間に何らかの参照関係がある場合

- 2つの動作の間に何らかの関係がある場合

ラベルの一覧を表2に掲げた。以下では、これらを順に説明する。

#### 3.3.1 主語(ガ格, subj)・対象(ヲ格, d-obj)・方向(ニ格, i-obj)

主語(ガ格)・対象(ヲ格)・方向(ニ格)は、調理者の動作とその対象食材を関連付ける。

主語(ガ格)は多くの場合、『食材(F)』と『食材の動作(Af)』を関係づける。たとえば、「/水/F が /沸騰/Af したら」という文は次のような枝を生成する。

例) /水/F  $\xrightarrow{\text{subj}}$  /沸騰/Af

対象(ヲ格)は、『調理者の動作(Af)』とその対象となる『食材(F)』とを関係づける。たとえば、「/チーズ/F を /おろ/Ac しておいて」という文は次のような枝を生成する。

例) /チーズ/F  $\xrightarrow{\text{d-obj}}$  /おろ/Ac

それ以前の調理動作によって生成された食材を対象として調理動作を行う場合は、その対象食材が文中に表れていない場合がある。たとえば、「玉葱を炒めたあと取り出す」とある場合、「/取り出/Ac」した対象の食材は、「/玉葱/F を /炒め/Ac」た結果として生成された食材である。このように、ある動作の前の食材と後の食材を別のものとして区別し、後の参照においては、どちらであるかを明示する。

例) /炒め/Ac  $\xrightarrow{\text{d-obj}}$  /取り出/Ac

方向(ニ格)は、食材を『食材(F)』に混ぜる場合や、食材を『容器(T)』に入れる場合など、主に調理動作と対象食材の行き先を関係づける。たとえば「/材料/F を /ボール/T に /入れ/Ac てる」という文は以下のようなようになる。

例) /材料/F  $\xrightarrow{\text{d-obj}}$  /入れ/Ac

例) /ボール/T  $\xrightarrow{\text{i-obj}}$  /入れ/Ac

#### 3.3.2 道具デ格(T-comp)・食材デ格(F-comp)

道具デ格は、『調理者の動作(Ac)』と、それを実行するのに使用される『道具(T)』を関連づける。たとえば「アルミホイルで覆い」という文は以下のようなようになる。

例) /アルミホイル/T  $\xrightarrow{\text{T-comp}}$  /覆/Ac

また、『調理者の動作(Ac)』を実行する道具が食材である場合もある。これは食材デ格で指定する。たとえば、「/オリーブオイル/F で /焼/Ac きます」という文は以下のようなようになる。

例) /オリーブオイル/F  $\xrightarrow{\text{F-comp}}$  /焼/Ac

#### 3.3.3 同一の食材(F-eq)・食材の部分(F-part-of)・食材の集合(F-set)

すでに登場した食材と同一の物体を指している固有表現を関連付けるためにいくつかのラベルを用意した。たとえば「1) 玉ねぎを切ります、2) 1) を炒めます」の場合、2個目の「1)」は「手順1)」の結果(すなわち切った結果)と同一食材なので、手順1)の最後の動作を「同一の食材(F-eq)」とする。

例) /切/Ac  $\xrightarrow{\text{F-eq}}$  /1)/F

以前に登場した食材の一部を指していると判断できる場合、たとえば「1) 玉ねぎの皮を剥いて」という文においては、「皮」は「玉ねぎ」の一部であると考えられるため、「皮」を「玉ねぎ」の『食材の一部(F-part-of)』に指定する。

例) /皮/F  $\xrightarrow{\text{F-part-of}}$  /玉ねぎ/F

以前に登場した食材の集合を指していると判断できる場合、たとえば「1) 玉ねぎと人参を切ります。2) 野菜を炒めます」という文章では、「野菜」が「玉ねぎと人参」を指していると考えられるため、「野菜」の「食材の集合(F-set)」に「玉ねぎ」と「人参」を指定する。

例) /玉ねぎ/F  $\xrightarrow{\text{F-set}}$  /野菜/F

例) /人参/F  $\xrightarrow{\text{F-set}}$  /野菜/F

#### 3.3.4 同一の道具(T-eq)・道具の一部(T-part-of)

すでに登場した道具と同一の物体を指している単語同志を関連付ける。たとえば「蓋をして10分煮込みます。蓋を外してさらに煮詰めます」の場合、蓋はどちらも同一物体と考えられるので、以下のようにする。

例) /蓋/T  $\xrightarrow{\text{T-eq}}$  /蓋/T

表 3 フローグラフコーパスの 1 レシピあたりの諸元

出典	文数	レシピ固有表現数 (葉の数, 中間ノード数)	単語数	文字数
無作為 (200 レシピ)	6.51	36.34 ( 13.91, 22.43)	118.65	180.51
肉じゃが (66 レシピ)	6.91	48.05 ( 22.12, 25.92)	132.95	198.70

以前に登場した道具の一部を指していると判断できる場合、たとえば「圧力鍋で煮ます。10分たったら蓋を外し」の場合、「蓋」は「圧力鍋」の蓋であると考えられるため、「蓋」の『道具の一部 (T-part-of)』に「圧力鍋」を指定する。

例)  $\boxed{/圧力鍋/T} \xrightarrow{\text{T-part-of}} \boxed{/蓋/T}$

### 3.3.5 同一の動作 (A-eq)

実際には一度しか行わない動作が、文中では複数回記述される場合がある。たとえば「玉葱<sub>1</sub>をみじん切り<sub>2</sub>します。先ほど切<sub>3</sub>った玉葱<sub>4</sub>を」の場合、後者の「切<sub>3</sub>」は前者の「みじん切り<sub>2</sub>」を指示しているにすぎず、実際には行わなくてよい。このような場合に、両固有表現を『同一の動作 (A-eq)』によって関連づけ、実際に行う動作と行わない動作を区別する。この例の場合には、共参照も含めて以下のような部分グラフになる。

例)  $\boxed{/玉葱_1/F} \xrightarrow{\text{d-obj}} \boxed{/みじん 切り_2/Ac}$   
 $\boxed{/玉葱_4/F} \xrightarrow{\text{d-obj}} \boxed{/切_3/Ac}$   
 $\boxed{/みじん 切り_2/Ac} \xrightarrow{\text{A-eq}} \boxed{/切_3/Ac}$   
 $\boxed{/玉葱_1/F} \xrightarrow{\text{F-eq}} \boxed{/玉葱_4/F}$

上述の関係は『調理者の動作 (Ac)』だけでなく、『食材の動作 (Af)』にも発生する。たとえば『水を沸騰させる。沸騰したら...』の場合、以下のようになる。

例)  $\boxed{/沸騰 させ/Ac} \xrightarrow{\text{A-eq}} \boxed{/沸騰/Af}$

### 3.3.6 動作を行うタイミング (V-tm)

動作を行うタイミングは調理を行う上で重要な指示情報であり、「したら」「するまで」「しないように」といった記述で表現される。たとえば「煮立ってきたら灰汁を取り」の場合は以下のようにする。

例)  $\boxed{/煮立/Af} \xrightarrow{\text{V-tm}} \boxed{/取/Ac}$

### 3.3.7 その他の修飾語句 (other-mod)

その他のすべての修飾関係はこの関係によって記述される。たとえば次のようなものがある。

例)  $\boxed{/四つ切/Sf} \xrightarrow{\text{other-mod}} \boxed{/する/Ac}$

例)  $\boxed{/45 分/D} \xrightarrow{\text{other-mod}} \boxed{/焼/Ac}$

## 3.4 コーパス

コーパスの統計情報を表 3 に示す。Mori ら [3] は Cookpad<sup>注1</sup> からレシピをランダムに選び、上述の基準に沿ってアノテーションを人手で行いコーパスを構築している。これらのランダムに選ばれたレシピにはお菓子や肉じゃがの作り方な

ど多様なレシピが含まれているため、レシピテキストの記述の偏りは少ない。

## 4. フローグラフの生成

この章では我々の提案するフローグラフの生成法を説明する。我々の手法は、Mori ら [8] などの方法で、固有表現が抽出されたグラフの頂点がすでに得られていると仮定する。この状態で、辺を生成した後にラベリングするという 2 段階の処理でフローグラフである DAG を生成する。

フローグラフの辺の生成は、MST を使用して木を生成した後に、さらに有用な辺を追加するという方法を取る。通常 MST は係り受け解析に使われており、対象となるのは 1 文のみである。今回我々は MST を使った解析を、1 文ではなく、複数の文からなるレシピテキスト全体に適用し、グラフを生成する。

第 2 節で述べたように、MST は大きなグラフであっても遷移ベースのアプローチより頑健であるという特徴を持つ。我々は文章全体を解析するので、頂点の数は一般の係り受け解析の場合よりも多くなると考えられる。したがって、グラフの大きさに対して頑健なグラフベースのアプローチを採用する。

### 4.1 MST による木の生成

MST による解析では、辺のスコアの合計を最大化させるような木を解析結果として出力する。レシピテキストから抽出された頂点の集合を  $V = \{v_1, v_2, \dots, v_{|V|}\}$ 、 $V$  より生成される全域木全体の集合を  $\mathcal{G}$  とする。さらに、2 つの頂点  $v_i$  と  $v_j$  を受け取って実数値のスコアを返す関数  $Score(v_i, v_j)$  の存在を仮定する。このスコア関数は、 $v_i$  を始点とし  $v_j$  を終点とする有向辺を作る場合の尤度を表現している。このようにしてありうる全ての辺のスコアリングをした状態で、MST を使った木の生成は、辺のスコアの合計値を最大にするような全域木を求める問題として以下のように定式化する:

$$\hat{G} = \operatorname{argmax}_{G \in \mathcal{G}} \left\{ \sum_{(v_i, v_j) \in G} Score(v_i, v_j) \right\}$$

この問題は Chu-Liu-Edmonds のアルゴリズム [15] [16] を使えば解けることが知られている。

我々はスコア関数  $Score(v_i, v_j)$  を確率を使って次のように定義する。

$$Score(v_i, v_j) = \frac{\exp\{\Theta \cdot feat(v_i, v_j)\}}{\sum_{w \in V \setminus \{v_i\}} \exp\{\Theta \cdot feat(v_i, w)\}} \quad (1)$$

ここで、 $V^2 = V \times V$  であり、 $\Theta$  はあらかじめ学習された高次元の重みベクトルであり、 $feat(v_i, v_j)$  は辺を高次元の特徴空間に写像する関数である。スコア関数は頂点  $v_i$  から頂点  $v_j$  に辺を張る時の確率を周囲の単語、品詞、読み、固有表現、固有表現タグなどから計算する。

(注1): <http://cookpad.com/> 2014 年 2 月 10 日アクセス。

```

1:  $G \leftarrow V$  の最大全域木
2:  $A \leftarrow$  無閉路のまま  $G$  に追加できる辺の系列
3:  $A$  を  $Score$  に従って値の大きな順番にソート
4:  $n \leftarrow 0$ 
5: for  $(v, w) \in A$  do
6:   if  $Score(v, w) > Penalty(n)$  then
7:      $G \leftarrow G + \{(v, w)\}$ 
8:      $n \leftarrow n + 1$ 
9:   end if
10: end for
11: return  $G$ 

```

図 4 DAG 生成のアルゴリズム

我々は対数線形モデル [17] を使い、正しい辺のスコアができるだけ高くなるように、パラメータ  $\theta$  を学習する。 $\{(V_t, v_t, w_t)\}_{t=1}^T$  を  $T$  インスタンスの学習セットとする。ここで、 $V_t$  は頂点の集合、 $(v_t, w_t)$  は正解の辺である。このような学習セットが与えられたとき、パラメータ推定は以下の尤度関数を最大化するように行う:

$$\sum_{t=1}^T \left\{ \log \frac{\exp(\theta \cdot feat(v_t, w_t))}{\sum_{w \in V_t \setminus \{v_t\}} \exp(\theta \cdot feat(v_t, w))} \right\} - \frac{C}{2} \|\theta\|^2$$

今回我々は DAG を扱っているため、1 つの頂点から辺が 2 本以上出ることがある。すなわち、 $V$  を頂点の集合、 $v$  を始点とすると、 $w$  と  $w'$  という 2 つ頂点が  $v$  の終点として学習データに存在する可能性がある。このような場合、我々は  $(V, v, w)$  と  $(V, v, w')$  の両方を学習インスタンスとして追加する。係り受け解析のように木を扱っている場合は、終点が 2 つあるような始点は存在しない。

#### 4.2 DAG の生成

DAG の生成は、図 4 に示すように、MST を用いて木を生成した後、無閉路のまま追加できる辺をさらに追加することで行う。図 4 において、 $Score$  は MST を求める時に使用する辺のスコア関数であり、 $Penalty(n)$  は  $n$  本目の辺を追加する場合のペナルティを返す関数である。アルゴリズムから分かるように、辺のスコアがペナルティを超えた場合に辺は追加される。ペナルティ関数によって辺の追加が制限されつつ、MST では拾えなかったスコアの高い辺を追加することができる。

我々はペナルティ関数として以下の式を用いる:

$$Penalty(n) = \frac{\xi}{\lambda e^{-\lambda n}}$$

右辺の分母はパラメータを  $\lambda$  とする指数分布であり、分子の  $\xi$  はスケールパラメータである。

ペナルティ関数の一部に指数分布を使うのは、木から DAG になるときの辺の追加具合が指数分布で表現できると考えられるからである。我々は表 3 のコーパスを使って、フローグラフの辺の数が、フローグラフの部分木の辺の数よりもどれくらい多いのかを計算した。図 5 は、横軸に、レシピフローグラフの辺が同じフローグラフから作られる部分木の辺の数よりもいく

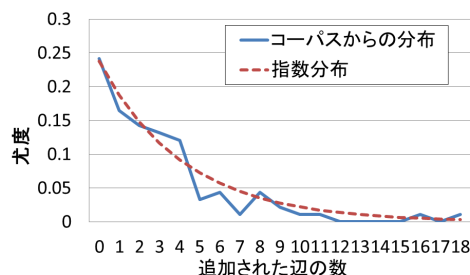


図 5 コーパスからの分布と指数分布の比較

つ多いかを取り、縦軸に、その数だけ辺が追加されたレシピの数を全レシピ数で割って正規化したグラフである。このグラフは、コーパスから推定した、辺の追加に対する尤度を表現する。図 5 から、指数分布は辺の追加具合をよくモデリングしていることが分かる。さらに、辺の追加数の尤度に指数分布を使うことによって、いかなる辺の数に対してもその尤度を与えることができる。

$\xi$  の最適な値は MST におけるスコア関数に依存する。具体的な値は、削除補間と同様の方法を用いて推定する。

#### 4.3 辺のラベリング

前節までの手法で辺を生成した後に、各辺にラベルを付けてフローグラフを出力する。我々は、辺のラベリングの問題を多クラス分類問題と見なし、サポートベクタマシン (SVM) を使用してラベリングを行う。すなわち、入力グラフの各辺に対し、辺の素性から最適な辺のラベルを選ぶ問題と設定する。

#### 4.4 素性

MST でのスコア関数の計算と辺のラベリング時に使用する素性を説明する。どちらの場合であっても、素性は 2 つの頂点  $v_i$  と  $v_j$  の間に辺を張った場合の特徴を高次元の特徴ベクトルとして表現する。

単独の素性は以下の通りである。

- 符号を含めた固有表現間の距離  $j - i$
- $v_i$  と  $v_j$  が同じ文にあるか
- $v_i$  と  $v_j$  が同じ手順にあるか
- 各  $v_i$  と  $v_j$  の表記、固有表現タグ
- 各  $v_i$  と  $v_j$  の前後を含む 3 単語

さらに組み合わせ素性を下に示す。

- $v_i$  の固有表現タグ  $\wedge$   $v_j$  の固有表現タグ  $\wedge$   $v_i$  と  $v_j$  の間に動作があるか  $\wedge$   $v_i$  と  $v_j$  が同じ手順にあるか
- $v_i$  の固有表現タグ  $\wedge$   $v_j$  の固有表現タグ  $\wedge$   $v_i$  と  $v_j$  の間に動作があるか  $\wedge$   $v_i$  と  $v_j$  の間に助詞があるか
- $v_i$  の固有表現タグ  $\wedge$   $v_j$  の固有表現タグ  $\wedge$   $v_i$  と  $v_j$  が同じ手順にあるか  $\wedge$   $v_i$  の直後の動作
- $v_i$  の固有表現タグ  $\wedge$   $v_j$  の固有表現タグ  $\wedge$   $v_i$  と  $v_j$  の間の助詞
- $v_i$  の表記  $\wedge$   $v_j$  の表記  $\wedge$   $v_j$  の固有表現タグ  $\wedge$   $v_i$  と  $v_j$  が同じ手順にあるか

ここで、 $\wedge$  は素性の連結を表す。

表 4 フローグラフの生成実験の結果

手法	適合率	再現率	F 値
ルールベース (ラベルなし)	75.9	69.4	72.5
提案手法 (ラベルなし)	80.9	75.8	78.3
提案手法 (ラベルあり)	72.4	67.8	70.0

## 5. 評価実験

提案法の有効性を示すために我々はレシピテキストから DAG を生成する実験を行った。この節では実験方法を説明し、結果を評価する。

### 5.1 実験設定

実験には第 3 節で説明したフローグラフコーパスを実験に使用した。以下の実験では、コーパスの固有表現をそのまま使用して、辺の生成を行った。すなわち、グラフにおけるノードはすべて正解のノードである。また、実験は 10 分割交差検定によって行った。

ベースラインとして、Hamada ら [10] のルールに基づく手法を我々のフローグラフの定義に読み直して実験した。Hamada らのグラフの設計は我々のそれとは異なっている。この手法を、我々のグラフの定義で置き換えると、食材や調理器具などのノードから、それらの直後の動作ノードに辺を張るということになる。この手法では辺が左から右へ張られるので、DAG ではなく木が生成される。また、辺のラベリングを行っていないため、このベースラインとの比較は辺の生成までである。

評価基準は適合率と再現率の調和平均である F 値を使用し、これらの値を辺の数に対して計算した。解析器が出力した辺の数を  $N_{sys}$ 、正解の辺の数を  $N_{ref}$ 、解析器の出力した辺と正解の辺の共通の部分の数を  $N_{int}$  と書くと、適合率、再現率および F 値はそれぞれ以下のように計算される。

$$\text{適合率} = \frac{N_{int}}{N_{sys}}$$

$$\text{再現率} = \frac{N_{int}}{N_{ref}}$$

$$\text{F 値} = \frac{2N_{int}}{(N_{ref} + N_{sys})}$$

### 5.2 結果と考察

表 4 に、既存手法 (ルールベース; Hamada ら [10] の再実装) および提案手法によるグラフ形状の推定の精度、さらに、提案手法においてラベル推定を含めた場合の精度を示す。なお、既存手法ではラベルなしのグラフを提案しているので、ルールによるラベル推定は行っていない。グラフ形状の推定において、既存手法と提案手法を比べると、提案手法のほうが高い精度を実現していることがわかる。この結果から、我々の手法の有効性が確認できる。学習には 180 レシピが用いているが、これを増量することは比較的容易であることを考えると、機械学習に基づく方法は有望であるといえる。表から、ラベルの推定を含めると F 値は少々 10% 程度下がっていることがわかる。正解の辺に対しては、90% 程度の精度でラベル推定が行えていることを意味する。これらのことを考え合わせると、さらなる改善に

表 6 1 レシピあたりの辺のラベル数

辺のラベル	数
subj	2.15
d-obj	15.67
i-obj	7.22
F-comp	0.65
T-comp	1.32
F-eq	3.15
F-part-of	2.37
F-set	0.15
T-eq	0.44
T-part-of	0.39
A-eq	0.53
V-tm	1.06
other-mod	3.54
合計	38.62

はグラフの形状推定の精度を向上することが重要であると考えられる。

自然言語処理の観点から、各言語現象がどの程度解決できているかを明らかにするために、辺のラベルごとにラベル推定まで行った場合の詳細な結果を表 5 に示す。この表において、文内の列は辺の両頂点のレシピ固有表現が同一文内にある場合の再現率であり、文外はこれらが異なる文に出現している場合の再現率である。また、総合は文内と文外を合わせた場合の再現率である。

表 5 の subj、d-obj、i-obj は、言語現象としては、述語項構造に分類される。文外 i-obj を除いて、これらの精度は高い。つまり、調理動作や食材動作と対象および方向の関係が高い精度で推定できていることがわかる。ただし、方向がゼロ代名詞になって明示されていない場合は困難である。文外の subj の推定精度は非常に高いが、これは、subj が「～を煮ます。温まったら」などのように、高頻度の一定の表現パターンに対応するからと考えられる。

その一方で、F-eq、T/F-part-of や A-eq などの共参照に対応する辺の精度が著しく低いことが分かる。これは、コーパスの量が少ないためこれらの辺の学習が満足にできていないためであると考えられる。表 6 に 1 レシピあたりの辺のラベル数を示す。表 6 から分かるように、1 レシピあたりの F-eq、T/F-part-of、A-eq の数は 1 を下回っているものもあり、かなり少ないと言える。そこでこれらの精度を上げるには、調理ドメインの辞書を構築して素性に含めるなどをする必要があると考えられる。とくに F-eq は、具体的な食材名とその食材が属する集合の関係を記述しているため、上位と下位の間接関係を表す辺になっている。したがって、そのような関係を記述したオントロジー [18] などの外部知識を活用することで精度が上がると期待できる。

## 6. まとめ

本論文では、手続き文章をフローグラフ化するという自然言語処理の研究として、1 文を木に変換するという係り受け解析を応用し、文章全体をグラフに変換する手法を提案した。今回

表 5 フローグラフの推定結果における言語現象ごとの再現率

現象	ラベル	文内	文外	合計
述語項 構造	subj	84.2 ( 309/ 367)	93.5 ( 58/ 62)	85.5 ( 367/ 429)
	d-obj	90.4 (2684/2970)	54.3 ( 89/ 164)	88.5 (2773/3134)
	i-obj	72.9 ( 902/1238)	45.1 ( 93/ 206)	68.9 ( 995/1440)
	F-comp	78.1 ( 100/ 128)	0.0 ( 0/ 1)	77.5 ( 100/ 129)
	T-comp	78.5 ( 205/ 261)	0.0 ( 0/ 2)	77.9 ( 205/ 263)
小計		84.6 (4200/4964)	55.2 (240/ 435)	80.0 (4440/5399)
共参照, 照応	F-eq	8.1 ( 9/ 111)	23.2 (102/ 519)	17.6 ( 111/ 630)
	F-part-of	41.7 ( 125/ 300)	29.3 ( 51/ 174)	37.1 ( 176/ 474)
	F-set	0.0 ( 0/ 13)	5.9 ( 1/ 17)	3.3 ( 1/ 30)
	T-eq	33.3 ( 7/ 21)	4.5 ( 3/ 66)	11.5 ( 10/ 87)
	T-part-of	4.3 ( 2/ 47)	6.5 ( 2/ 31)	5.1 ( 4/ 78)
	A-eq	0.0 ( 0/ 6)	1.0 ( 1/ 99)	1.0 ( 1/ 105)
小計		28.7 ( 143/ 498)	17.7 (160/ 906)	21.6 ( 303/1404)
その他	V-tm	49.5 ( 104/ 210)	0.0 ( 0/ 2)	49.1 ( 104/ 212)
	other-mod	56.0 ( 385/ 687)	4.8 ( 1/ 21)	54.5 ( 386/ 708)
	小計	54.5 ( 489/ 897)	4.3 ( 1/ 23)	53.3 ( 490/ 920)
合計		76.0 (4832/6359)	27.3 (401/1364)	67.8 (5233/7723)

はレシピテキストを変換の対象とした。従来研究とは異なり、変換して出力されるフローグラフは DAG である。この DAG は、レシピテキストの固有表現をノードとし、述語項構造、共参照などの関係を辺とする。我々の提案手法による DAG の生成は、MST を使って木を生成したあと、さらに有用とおもわれる辺を追加するというものである。比較実験の結果、我々の手法は従来のルールによる木の生成よりも高い精度となった。

取扱説明書など、様々な手続き文書を検討した結果、頂点や辺のラベルは異なるものの、内容をフローグラフとして表現可能であることが分かっている。本論文で提案するレシピテキストのフローグラフへの変換手法は、同様の手続き文書にも適用可能である。提案手法は、調理レシピを含め様々な手順書の知的な検索や分析の実現に有用である。

#### 文 献

- [1] Liping Wang, Qing Li, Na Li, Guozhu Dong, and Yu Yang. Substructure similarity measurement in chinese recipes. In *Proceedings of the 17th international conference on World Wide Web*, pp. 978–988, 2008.
- [2] 山肩洋子, 角所考, 美濃導彦. 調理コンテンツの自動作成のためのレシピテキストと調理観測映像の対応付け. *電子情報通信学会論文誌*, Vol. J90-DII, No. 10, pp. 2817–2829, 2007.
- [3] 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*, 2014.
- [4] 森信介, 山肩洋子, 笹田鉄郎, 前田浩邦. レシピテキストのためのフローグラフの定義. *情報処理学会研究報告*, 第 NL214 巻, 2013.
- [5] 三浦宏一, 高野求, 浜田玲子, 井手一郎, 坂井修一, 田中英彦. 料理映像の構造解析による調理手順との対応付け. *電子情報通信学会論文誌*, Vol. J86-DII, No. 11, pp. 1647–1656, 2003.
- [6] Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pp. 523–530, 2005.
- [7] Joakim Nivre and Jens Nilsson. Pseudo-projective dependency parsing. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pp. 99–106, 2005.
- [8] Shinsuke Mori, Tetsuro Sasada, Yoko Yamakata, and Koichiro Yoshino. A machine learning approach to recipe text processing. In *Proceedings of Cooking with Computer workshop*, 2012.
- [9] Yoshio Momouchi. Control structures for actions in procedural texts and pt-chart. In *Proceedings of the Eighth International Conference on Computational Linguistics*, pp. 108–114, 1980.
- [10] Reiko Hamada, Ichiro Ide, Shuichi Sakai, and Hidehiko Tanaka. Structural analysis of cooking preparation steps in japanese. In *Proceedings of the fifth international workshop on Information retrieval with Asian languages*, No. 8 in IRAL '00, pp. 157–164, 2000.
- [11] Ryan McDonald and Fernando Pereira. Online learning of approximate dependency parsing algorithms. In *Proceedings of the Eleventh European Chapter of the Association for Computational Linguistics*, pp. 81–88, 2006.
- [12] Kenji Sagae and Alon Lavie. A best-first probabilistic shift-reduce parser. In *Proceedings of the 21th International Conference on Computational Linguistics*, 2006.
- [13] Ryan McDonald and Joakim Nivre. Analyzing and integrating dependency parsers. *Computational Linguistics*, Vol. 37, No. 4, pp. 197–230, 2011.
- [14] Daniel Flannery, Yusuke Miyao, Graham Neubig, and Shinsuke Mori. A pointwise approach to training dependency parsers from partially annotated corpora. *Journal of Natural Language Processing*, Vol. 19, No. 3, 2012.
- [15] Yoeng-Jin Chu and Tseng-Hong Liu. On the shortest arborescence of a directed graph. *Science Sinica*, Vol. 14, pp. 1396–1400, 1965.
- [16] Jack Edmonds. Optimum branchings. *Journal Research of the National Bureau of Standards*, Vol. 71B, pp. 233–240, 1967.
- [17] Vincent J. Della Pietra Adam L. Berger, Stephen A. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, Vol. 22, No. 1, 1996.
- [18] 土居洋子, 辻田美穂, 難波英嗣, 竹澤寿幸, 角谷和俊. 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. *電子情報通信学会技術研究会報告*, 2014.