

手順文書からの意味構造抽出

Extracting Semantic Structure from Procedural Texts

前田 浩邦
Hirokuni Maeta

サイボウズ株式会社
Cybozu, Inc.
hirokuni.maeta@gmail.com

山肩 洋子
Yoko Yamakata

東京大学大学院 情報理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo
yamakata@hal.t.u-tokyo.ac.jp

森 信介
Shinsuke Mori

京都大学学術情報メディアセンター
Academic Center for Computing and Media Studies, Kyoto University
forest@i.kyoto-u.ac.jp

keywords: procedural text, recipe, semantic structure, natural language processing

Summary

We propose a method for extracting semantic structure from procedural texts for more intelligent search or analysis. Procedural texts represent a sequence of procedures to create an object or to make an object be in a certain state, and have many potential applications in artificial intelligence. Procedural texts are relatively clear without modality nor dependence on viewpoints, etc. Thus they can be described their procedures using flow graphs. We adopt recipe texts as procedural text examples and directed acyclic graphs (DAGs) to represent semantic structure. Nodes of a flow graph are important terms in a recipe text and vertices are relationships between the terms such as language phenomena including dependency, predicate-argument structure, and coreference. Because trees can not represent the procedures of recipes sufficiently, DAGs are adopted as the representation of recipes. We first apply word segmentation, automatic term recognition, and then convert the entire text into a flow graphs. For word segmentation and automatic term recognition, we adopt existing methods. Then we propose a flow graph estimation method from term recognition results. Our method is based on the maximum spanning tree algorithm, which is popular in dependency parsing, and simultaneously deals with language phenomena listed above. We experimentally evaluate our method on a flow graph corpus created from various recipe texts on the Internet.

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、対象を手順文書(手続きを記述した文書)に限定し、自動的に文書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。

文書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、本研究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順文書に限定する。手順文書とは、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。手順文書は、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

るのかということを記述している。手順文書を、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、動作と動作の関係が明確になる。

本論文では、具体的に処理する手順文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、述語項構造や共参照等の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、cookpad、楽天レシピといった多数のインターネットサイトがあり、数百万のレシピが掲載されている。また、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文書と考えられる。フローグラフを利用する先行研究として、レシピテキストのフローグラフ化が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 [Wang 08]。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応を自動で獲得する研究が行われている [山肩 07]。

本研究では、レシピと、その意味内容を表現したフローグラフからなるコーパス [森 13, Mori 14] を用いる。フ

フローグラフの頂点は、手順文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。レシピの構造化の初期の研究 [三浦 03] では、内容表現の形式として木が用いられている。しかし、オレンジの果肉と皮を分けてそれぞれ別の調理をする場合など、材料を分割することがあるため、手順文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。上述のレシピとフローグラフのコーパス [森 13, Mori 14] では、内容表現として最終生成物を根とする根つきの DAG が用いられているため、レシピの意味内容をより適切に表現することができる。したがって、本研究ではこのコーパスを採用する。

本論文では、レシピテキストをその意味表現であるフローグラフ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。文書理解を目的とする場合、例えば、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。まず Mori ら [Mori 12] が提案する単語分割、重要語の認識を行う。その後、本手法では、重要語の間に辺を張り、得られた各辺にラベルを付与することで、フローグラフを得る。辺を張る過程においては、係り受け解析、述語項構造、共参照などの言語現象を個別に扱うことなく重要語間の関係を抽出し、辺にラベルを付与する過程において、重要語間の関係が何であったかを推定する。

本論文で提案するフローグラフへの変換手法は、機械学習に基づいた係り受け解析 [McDonald 05] をレシピテキスト全体に拡張したものと見ることができる。機械学習に基づく提案手法の有効性を検証するために、従来のルールに基づいたフローグラフの推定手法 [Hamada 00] 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

レシピテキストのフローグラフ化には単語分割とレシピ用語の認識が必要である。Mori ら [Mori 12] 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とで、調理ドメインでの単語分割、レシピ用語の認識、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 [Momouchi 80] 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

を含む手順文書全般の解析手法を提案している。Hamada ら [Hamada 00] 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、この手法は料理の専門家が執筆し、編集者によって編集されたテキストに適用されることを前提としている。したがって、Web レシピの大半を占める、一般ユーザが執筆したレシピにあらわれる様々なスタイルの文書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 [山肩 07] 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した。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 章でも述べたように、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。Kiddon ら [Kiddon 15] は教師無し学習を用いてレシピテキストをフローグラフに変換している。しかし、この手法では共参照関係や動作の行う条件といった言語現象を扱っておらず、手順を実行するうえで必要な情報が欠落している。Jermurawong ら [Jermurawong 15] のフローグラフも同様の問題を持つ。

本論文では係り受け解析の手法を応用することでレシピ全体をグラフ化する。係り受け解析の手法には大きく分けて 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 [McDonald 05, McDonald 06] と遷移ベースのアプローチ [Nivre 05, Sagae 06] である。さらに、遷移ベースのアプローチによる DAG 推定の手法も存在する [Sagae 08, Wang 15]。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 [McDonald 05] 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 [McDonald 11]、日本語の係り受け解析にも導入されている [Flannery 12]。McDonald ら [McDonald 11] 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、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。レシピテキストのフローグラフ化では文書全体を解析するため、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、本論文では遷移ベースの係り受け解析や DAG 推定は採用せず、グラフの大きさに対して頑健な MST による手法を採用する。

1. 両手鍋で油を熱する。

セロリと青ねぎとニンニクを加え、1分ほど炒める。

2. ブイヨンと水とマカロニと胡椒を加えて、

パスタが柔らかくなるまで煮る。

3. 刻んだセージをまぶす。

表 1 レシピ用語の種類

クラス	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	

図 1 レシピ調理手順の例

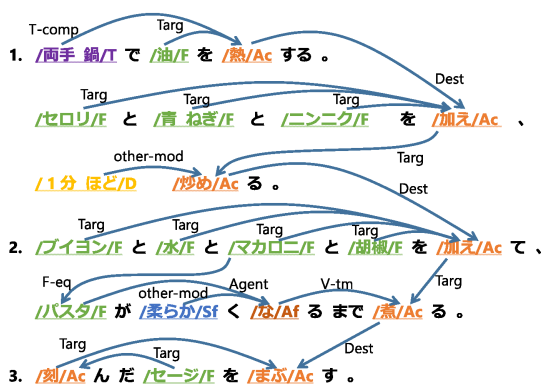


図 2 図 1 のレシピから変換されたフローグラフ

表 2 辺のラベル

ラベル	意味
Agent	主語
Targ	対象
Dest	方向
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

3. レシピフローグラフ

本章ではレシピフローグラフを概説する。詳しい定義は、文献 [笹田 15, 森 13, Mori 14] に述べられている。

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)

以上に加えて、調理手順の説明のための写真や完成した料理の写真、あるいは、レシピ作成者や利用者のコメントなどの情報が付与されている場合もあるが、本研究では扱うのは (3) のみである。レシピの調理手順文書の例を図 1 に、この調理手順に対応するフローグラフを図 2 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を切る必要があるが、ニンジンを茹でる前にジャガイモを切る必要はない。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、食材の分離である。^{*1} したがって、フローグラフは DAG である。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の頂点がある。これは、その手順で行われる最後の加工動作に対応し、その結果が料理の完成品であることを意味する。

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。このような単語列をレシピ用語と呼ぶ。表 1 に、レシピ用語の種類 (クラス) とその意味を掲げる。詳しいレシピ用語の定義は [笹田 15] に述べられている。

フローグラフの辺

文書中のレシピ用語間に以下で挙げる 3 つのうちいずれかの関係が認められる場合、それらのレシピ用語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。フローグラフの辺は、左から右に行くものだけでなく、右から左に行くものもある。

^{*1} 直感的には食材の分離が分かりやすいが、実際のレシピで類出する例は、図 2 におけるマカロニからパスタへの辺のような共参照等による分岐である。

- 述語項関係がある場合 (ゼロ照応も含む)
- 同種の 2 つのレシピ用語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

ラベルの一覧とその意味を表 2 に掲げた。フローグラフの辺やラベルの詳しい定義は [森 13, Mori 14] に述べられている。

3.2 コーパス

コーパスの統計情報を表 3 に示す。文献 [森 13, Mori 14] は cookpad^{*2} 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 [Mori 12] などの方法で、レシピ用語が抽出されグラフの頂点がすでに得られると仮定する。この状態で、辺を生成した後に辺のラベルを推定するという 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 章で述べたように、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。我々は文書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 [Chu 65, Edmonds 67] を使えば解けることが知られている。日本語の文書は左から右に係り受けが生じるため、アルゴリズムに入力するグラフの根を手順文書の一番最後のレシピ用語にしている。

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。

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

ここで、 Θ 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、レシピ用語、レシピ用語クラスなどから計算する。

我々是对数線形モデル [Berger 96] 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 Θ 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。

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

ここで C は正規化定数である。

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 つの頂点が 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 は、無閉路という条件を満たしたまま木に有向辺を加えたグラフであると見ることができる。この観点からレシピフローグラフにおける DAG と木の違いを調べるため、レシピフローグラフの辺が、同頂点数の木に対してどれだけ多いのかを調べた。図 3 の折れ線は、横軸に、レシピフローグラフの辺が同頂点数の木の辺の数よりもいくつ多いかを取り、縦軸に、レシピの割合を取るグラフである。この計算には表 3 のコーパスを使用し

*2 <http://cookpad.com/> 2015 年 11 月 25 日アクセス。

表3 フローグラフコーパスの1レシぴあたりの諸元

出典	文数	レシぴ用語数 (葉の数, 中間ノード数)	単語数	文字数
無作為 (200 レシぴ)	6.51	36.34 (13.91, 22.43)	118.65	180.51

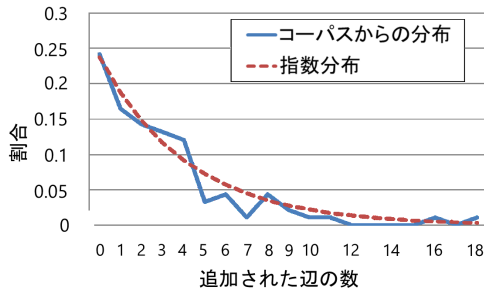


図3 コーパスからの分布と指数分布の比較

```

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:  $n \leftarrow 1$ 
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:   else
10:     break
11:   end if
12: end for
13: return  $G$ 
    
```

図4 DAG生成のアルゴリズム

た。このグラフは、コーパスから推定した、辺の追加に対する度数を表現する。図3の曲線(破線)は指数分布であり、辺の追加具合をよくモデリングしていることが分かる。

本手法では、DAGを生成するために、まずMSTを用いて木を生成した後、閉路を作ることなく追加できる辺をさらに追加する。アルゴリズムを図4に示す。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。上記の考察から、ペナルティは指数分布を用いて次のように定義する。

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

右辺の分母はパラメータを λ とする指数分布であり、分子の ξ はスケールパラメータである。 n が大きくなるほどペナルティは大きくなる。ペナルティ関数によって辺の追加が制限されつつ、MSTでは捨てられてしまったスコアの高い辺を追加することができる。

パラメータ (λ, ξ) は、学習コーパスに対して最適とな

るように求める。パラメータ λ は、学習コーパスでの辺の追加度数に対して2乗誤差が最小になるように決定する。 ξ は、パラメータ λ を推定した後にヘルドアウト法を用いて推定する。まず、学習コーパスを2つに分割し、片方をMSTのパラメータ θ の推定に用いる。次に、推定したパラメータ θ を用いてDAGの推定を残りのコーパスに対して行い、推定精度が一番高くなる ξ を計算する。

4.3 辺のラベル推定

前項までの手法で辺を生成した後、各辺にラベルを付けてフローグラフを出力する。我々は、辺のラベル推定を多クラス分類問題と見なし、サポートベクタマシン(SVM)を使用して推定する。すなわち、入力の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。なお、学習にはコーパスでの正解の辺を用いる。

4.4 素性

MSTでのスコア関数の計算と辺のラベル推定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。

- 符号を含めたレシぴ用語間の距離 $j - i$
- 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、レシぴ用語クラス
- 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以下に示す。

- v_i のレシぴ用語 \wedge v_j のレシぴ用語クラス \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 のレシぴ用語クラス \wedge v_j のレシぴ用語クラス \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 のレシぴ用語クラス \wedge v_j のレシぴ用語クラス \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 のレシぴ用語クラス \wedge v_j のレシぴ用語クラス \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 のレシぴ用語クラス \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。

5. 評価実験

提案法の有効性を示すためにレシぴ手順文書からDAGを生成する実験を行った。この章では実験方法を説明し、

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

結果を評価する。

5.1 実験設定

実験には第 3 章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、コーパスのレシピ用語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手順文書内に存在するすべての正しいレシピ用語である。

実験は 10 分割交差検定によって行った。まず、全コーパスを学習データとテストデータとして 9:1 に分割した組を 10 個作成した。ξ の推定においては、各々の学習・テストデータの組に対して学習データの 9/10 を MST のパラメータ推定、1/10 をヘルドアウトデータとして推定に用いた。したがって、MST のパラメータ推定、ヘルドアウト、テストのデータサイズの割合は 8.1:0.9:1.0 となっている。

ベースラインとして、Hamada ら [Hamada 00] 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、辺のラベル推定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。

$$\text{適合率} = \frac{N_{int}}{N_{sys}}, \quad \text{再現率} = \frac{N_{int}}{N_{ref}},$$

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

5.2 結果と考察

表 4 に、既存手法 (ルールベース; Hamada ら [Hamada 00] の再実装) および提案手法によるグラフ形状 (ラベルを含めないノード間の有向辺の有無) 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ため、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 レシピを用いているが、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、ラベルを含めたノード間の有向辺の有無の推定の F 値は 10% 程度下がっていることがわかる。存在する辺を正しく見つけられた場合、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。

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、辺のラベル推定まで行った場合でのラベルごとの詳細な結果を表 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用語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 Agent、Targ、Dest は、言語現象としては、述語項構造に分類される。文外の Dest 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 Agent の推定精度は非常に高いが、これは、Agent 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。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、上位と下位の間を表現する辺になっている。したがって、そのような関係を記述したオントロジー [土居 14] 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。すなわち、共参照に対応する辺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。

5.3 さらに精度の改善に関して

図 5 に、それぞれの辺ラベルに対する再現率と頻度のグラフをあげる。グラフの辺のラベルは、表 4 の言語現象ごとに楕円でグルーピングされている。点線は、線形回帰直線である。この図において、この直線の右下の方にあるラベルほど、出現頻度に対する推定精度がより低いといえる。したがって、F-part-of や F-eq のような共参

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	Agent	84.2 (309/ 367)	93.5 (58/ 62)	85.5 (367/ 429)
	Targ	90.4 (2684/2970)	54.3 (89/ 164)	88.5 (2773/3134)
	Dest	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)

表6 1 レシピあたりのラベルごとの辺の数

辺のラベル	数
Agent	2.15
Targ	15.67
Dest	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

照、照応の推定精度に改善の余地が多分にあるといえる。

Targ、Dest、及び other-mod は最も出現頻度の高いため、全体の精度に最も影響を与えている。この3つは回帰直線上にあるものの、表6から文外での Targ と Dest の推定精度が低いということが分かる。文外での Targ と Dest は、主に、動作 v_1 (Ac) による生成物を別の動作 v_2 (Ac) に接続する。その時、辺が Targ で接続するのか、もしくは Dest で接続するのかは、 v_1 に依存する。もし v_1 による生成物が食材 (F) の傾向を持つならば (例えば/切/Ac など)、 v_1 は Targ によって v_2 と接続する。また、 v_1 の結果が食器 (T) の傾向を持つならば (例えば/加/Ac など)、 v_1 と v_2 は Dest によって接続する。このことは、格フレー

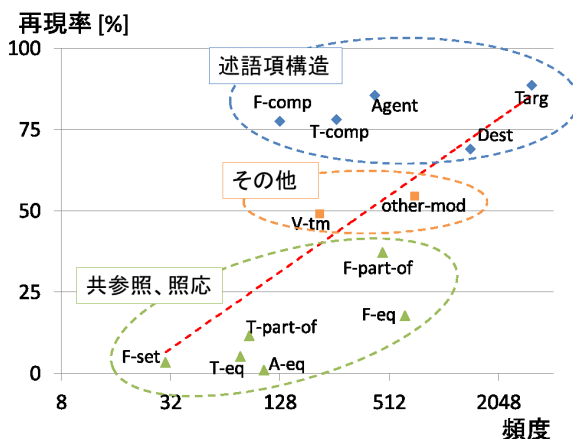


図5 頻度と再現率の関係

ム [Baker 98] に加えて、動作 (動詞) の結果どのような生成物ができる傾向にあるかという世界知識が、手順文書の解析に必要なことを意味している。

6. ま と め

本論文では、手順文書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、文書全体をグラフに変換する手法を提案した。今回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、レシピテキストのレシピ用語を頂点とし、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、さらに有用と推定さ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、レシピ以外の様々な手順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、他の手順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文書の知的な検索や分析の実現に有用である。

謝 辞

本研究は JSPS 科研費 26280084 および 24240030 の助成を受けたものです。ここに謝意を表します。

◇ 参 考 文 献 ◇

- [Baker 98] Baker, C. F., Fillmore, C. J., and Lowe, J. B.: The Berkeley FrameNet Project, in *Proceedings of the 17th International Conference on Computational Linguistics*, pp. 86–90 (1998)
- [Berger 96] Berger, A. L., Pietra, S. A. D., and Pietra, V. J. D.: A Maximum Entropy Approach to Natural Language Processing, *Computational Linguistics*, Vol. 22, No. 1, pp. 39–71 (1996)
- [Chu 65] Chu, Y.-J. and Liu, T.-H.: On the Shortest Arborescence of a Directed Graph, *Science Sinica*, Vol. 14, pp. 1396–1400 (1965)
- [土居 14] 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報告, 第 113 巻, pp. 37–42 (2014)
- [Edmonds 67] Edmonds, J.: Optimum Branchings, *Journal Research of the National Bureau of Standards*, Vol. 71B, pp. 233–240 (1967)
- [Flannery 12] Flannery, D., Miyao, Y., Neubig, G., and Mori, S.: A Pointwise Approach to Training Dependency Parsers from Partially Annotated Corpora, Vol. 19, No. 3, pp. 167–191 (2012)
- [Hamada 00] Hamada, R., Ide, I., Sakai, S., and Tanaka, H.: Structural Analysis of Cooking Preparation Steps in Japanese, in *Proceedings of the 5th International Workshop on Information Retrieval with Asian Languages*, No. 8 in IRAL '00, pp. 157–164 (2000)
- [Jermurawong 15] Jermurawong, J. and Habash, N.: Predicting the Structure of Cooking Recipes, in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 781–786, Lisbon, Portugal (2015), Association for Computational Linguistics
- [Kiddon 15] Kiddon, C., Ponnuraj, G. T., Zettlemoyer, L., and Choi, Y.: Mise en Place: Unsupervised Interpretation of Instructional Recipes, in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 982–992, Lisbon, Portugal (2015), Association for Computational Linguistics
- [McDonald 05] McDonald, R., Pereira, F., Ribarov, K., and Hajič, J.: 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)
- [McDonald 06] McDonald, R. and Pereira, F.: Online Learning of Approximate Dependency Parsing Algorithms, in *Proceedings of the 11th European Chapter of the Association for Computational Linguistics*, pp. 81–88 (2006)
- [McDonald 11] McDonald, R. and Nivre, J.: Analyzing and Integrating Dependency Parsers, *Computational Linguistics*, Vol. 37, No. 4, pp. 197–230 (2011)
- [三浦 03] 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)
- [Momouchi 80] Momouchi, Y.: Control Structures for Actions in Procedural Texts and PT-Chart, in *Proceedings of the 8th International Conference on Computational Linguistics*, pp. 108–114 (1980)
- [Mori 12] Mori, S., Sasada, T., Yamakata, Y., and Yoshino, K.: A Machine Learning Approach to Recipe Text Processing, in *Proceedings of Cooking with Computer Workshop* (2012)
- [森 13] 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)
- [Mori 14] Mori, S., Maeta, H., Yamakata, Y., and Sasada, T.: Flow Graph Corpus from Recipe Texts, in *Proceedings of the 9th International Conference on Language Resources and Evaluation* (2014)
- [Nivre 05] Nivre, J. and Nilsson, J.: Pseudo-Projective Dependency Parsing, in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pp. 99–106 (2005)
- [Sagae 06] Sagae, K. and Lavie, A.: A Best-First Probabilistic Shift-Reduce Parser, in *Proceedings of the 21st International Conference on Computational Linguistics* (2006)
- [Sagae 08] Sagae, K. and Tsujii, J.: Shift-Reduce Dependency DAG Parsing, in *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pp. 753–760, Manchester, UK (2008), Coling 2008 Organizing Committee
- [笹田 15] 笹田 鉄郎, 森 信介, 山肩 洋子, 前田 浩邦, 河原 達也: レシピ用語の定義とその自動認識のためのタグ付与コーパスの構築, 自然言語処理, Vol. 22, No. 2, pp. 107–131 (2015)
- [Wang 08] Wang, L., Li, Q., Li, N., Dong, G., and Yang, Y.: Substructure Similarity Measurement in Chinese Recipes, in *Proceedings of the 17th International Conference on World Wide Web*, pp. 978–988 (2008)
- [Wang 15] Wang, C., Xue, N., and Pradhan, S.: A Transition-based Algorithm for AMR Parsing, in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 366–375, Denver, Colorado (2015), Association for Computational Linguistics
- [山肩 07] 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)

〔担当委員: 岡崎 直観〕

2016 年 2 月 9 日 受理

著 者 紹 介



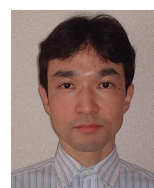
前田 浩邦

2012 年京都大学理学部卒業。2014 年同大学院情報学研究科知能情報学専攻修士課程修了。同年、サイボウズ株式会社入社。以来、グループウェアの開発に従事。



山肩 洋子 (正会員)

2005 年京都大学大学院情報学研究科博士後期課程研究指導認定退学。同年、京都大学研究員。2006 年情報通信研究機構専攻研究員。2010 年京都大学特定講師、准教授、特定准教授を経て、2015 年より日本学術振興会 特別研究員。2016 年より英国サセックス大学客員研究員。レシピや調理行動に関するマルチメディア処理の研究に従事。電子情報通信学会会員。博士 (情報学)。



森 信介

1998 年京都大学大学院工学研究科電子通信工学専攻修士後期課程修了。同年、日本アイ・ピー・エム株式会社入社。2007 年より京都大学学術情報メディアセンター准教授、2016 年同教授。京都大学博士 (工学)。1997 年情報処理学会山下記念研究賞受賞。2010 年、2013 年情報処理学会論文賞受賞。2010 年第 58 回電気科学技術奨励賞受賞。言語処理学会、情報処理学会、日本データベース学会、ACL 各会員。