投稿レシピの材料表における文字の隣接強度をもとにした助数詞および計量器の抽出

但馬康宏 [1]
[1] 岡山県立大学
tajima@cse.oka-pu.ac.jp

概要

本研究では，投稿型レシピデータの材料表について，そこに出現する助数詞および計量器の抽出を行う．一般にレシピの材料表に出現する語句は述語を含まず，文章となっていない場合がほとんどであり，既存の形態素解析手法では正しく語句の理解ができない．しかし，数値と助数詞以外にも計量器や注意書きなど自然言語の出現が避けられず語句の解析が必要である．本研究では出現する文字列の接合強度を定義し，語句の出現回数から求めることにより，辞書を用いずに解析を行う．本手法による解析で，助数詞および計量器に対しておよそ 8 割程度の正答率を得た．

1 はじめに

近年のウェブサービスにおいてレシピ情報とそのコンテンツ利用は非常に盛んであり，食に関する関心の高まりとともにその重要性は今後も増加していくものと思われる．一方，投稿情報によるデータの集積という性質上，統一的な記述ができず，情報の加工，知識の抽出という観点からはむずかしい問題が多い．レシピ情報の加工に関しては，グラフとして調理過程を扱うもの [1]，オントロジーの作成 [2] など様々な観点から研究が行われている．本研究では，材料表に出現する助数詞と計量器を辞書を使わずに抽出する．これは栄養価の計算では必須の処理であり，食材の正しい分量を理解することは重要な課題である．辞書を使わない言語処理は未知語の抽出 [3] やキーワード抽出 [4] などがあるが必ずしも多くない．

本研究では出現する文字列の接合強度を定義し，語句の出現回数から求めることにより，辞書を用いずに助数詞および計量器の抽出を行う．その結果，助数詞で 9 割，計量器 7 割程度の正答率を得た．

2 材料表の構成と分量表現

本研究では，クックパッドより提供されているデータ [5] を用いた．材料表は表 1 の形式である．食材名は材料の名前であり，同じ食材に関しても投稿者により多様な表現が存在する．本研究で注目するのは，分量の部分であり，表の例のとおり数値に助数詞がつく形が基本であるが，その前後に計量器や修飾語が付く場合が多い．また，数値を含まず「少々」「お好みで」などの語句のみの場合もある．

表 1 レシピにおける材料表

食材名	分量
にんじん	2 本
ピーマン	4 つ
れんこん	小 1
たまねぎ	1 つ
砂糖	小さじ 1 くらい
豚肉	ひき肉 250g
コショウ	少々
ニンニク	お好みで
醤油	適量（味見しながらね）
みりん	大さじ 3 くらい

本研究では材料表の分量表現について，数値が出現する項目の助数詞および計量器の抽出を行う．

3 助数詞および計量器の表現

分量表現のうち数値が出現するものに限ると，助数詞は数値の直後に出現し，計量器は数値の直前に出現することが多い．しかし助数詞にはその後に修飾語が付き「2 つくらい」や「3 本ほど」のようになる場合があり，数値の前の計量器には前後に修飾語が付き「大さじ山盛り」や「ほぼ小さじ」などとなる場合がある．これらの修飾語は複数付く場合もあり，形態素解析のような分割が必要となる．しかし，文章として記述されてはいないため一般の形態

素解析器などでは誤分割が多くなる．形態素解析器自体を分量表現に適した辞書と学習により構成することも考えられるが，正解データの準備など多大な労力を必要とする．

そこで本研究では，出現文字の統計情報から形態素の切れ目を推定するアルゴリズムを提案し，分量表現に対して効果的であることを示す．具体的には，文字列の中の文字間の接続強度を定義し，その値が文字列全体で偏りがなく設定できる位置を形態素の切れ目とし，分割を行う．

まず，助数詞および計量器の含まれる文字列として，分量表現中に出現する最初の数値に着目し，その数値の直前に出現するかな漢字文字列を w_p とし，直後に出現するかな漢字文字列を w_s とする．以後，w はこの文字列 w_p もしくは w_s を表すものとし，w を分割することを考える．

4 隣接強度に注目した形態素分割

図 1 および本節にて本研究における隣接強度の算出方法を示す．

4.1 データと出現確率の定義

分割を行いたい文字列を w とし，以後 c_0 から c_{n-1} の n 文字から成るとする $(w = c_0c_2\cdots c_{n-1})$．また，文字列を先頭から m 文字目を後半の先頭として，前半と後半に分割する様子を $[0, m, n]$ として表す．例えば $w = c_0c_2\cdots c_{n-1}$ を先頭から 2 文字までを前半 (c_1c_2) とし，それ以後を後半 $(c_3c_4\cdots c_{n-1})$ として分割する場合は $[0, 3, n]$ と表現される．複数の分割の場合も同様に，各分割の先頭の文字がもとの文字列の何文字目であるかを並べたものとする．例えば $w = c_0c_2\cdots c_{n-1}$ を 2 文字ごとに分割する場合は，$[0, 2, 4, \cdots, 2\lfloor\frac{n-1}{2}\rfloor, n]$ と表現される．さらに，分割 $[0, m_1, m_2, \cdots, m_k, n]$ を分割された部分文字列で表現する場合は $(v_0, v_1, \cdots, v_k)$ のように表現する．ここで v_i は $c_{m_i}c_{m_i+1}\cdots c_{m_{i+1}-1}$ となる文字列である．

分割対象文字列 w を集めたデータを W と表す．これには同じ文字列が複数回出現することもある．本研究では W に含まれる文字列の部分文字列の統計情報を利用する．データ W における w の出現数を D_w と表す．すべての $w \in W$ について，その部分文字列のすべてを集めたものを V とする．V にも頻出の部分文字列は複数回出現するものとする．文字列 $v \in V$ の V における出現数を C_w と表す．

文字列 $v \in V$ について，v の出現確率を

$$P(v) = \frac{C_v}{|V|}$$

と定め，$x \notin V$ である文字列 x については $P(x) = 0$ とする．

文字列 u についてその長さを $|u|$ とする．ある $w = c_1c_2\cdots c_n \in W$ を仮定し，u が w の部分文字列として出現する回数を $k_{u,w}$ とする．すなわち，$k_{u,w} = |\{i | c_ic_{i+1}c_{i+|u|} = w\}|$ である．さらに $k_u = \sum_{w\in W} k_{w,u}$ とする．これはデータ全体で文字列 u が出現する回数である．

ふたつの文字列 u, v について，隣接出現確率を以下のように定める．

$$P(u, v) = \frac{k_{uv}}{\sum_{x\in V} k_{ux}}$$

これは，すべての部分文字列の隣接回数を分母とし，u と v が隣接して出現する回数の割合である．

4.2 隣接強度の定義

分割対象文字列 $w \in W$ およびその分割 $(v_0, v_1, \cdots, v_k)$ について，v_i と v_j $(0 \le i, j \le k)$ の隣接強度を以下のように定義する．

$$\begin{aligned} score(v_i, v_j) &= |v_i|P(v_i)\cdot|v_j|P(v_j) \\ &\quad\cdot\sum_{u\in V} P(u)\left(\frac{2P(v_i, u)P(u, v_j)}{P(v_i, u) + P(u, v_j)}\right) \end{aligned}$$

これは 1 行目の出現確率の積で，分割位置の前後の出現文字列 v_i, v_j が形態素として正しい分割であるかを評価している．すなわち，それぞれの出現確率が高ければその分割は正しい分割となっている可能性が高く，まちがった位置で分割した場合は v_i か v_j があまり出現しない妙な文字列となっている可能性が高いためである．また 2 行目の総和の項は v_i の後ろおよび v_j の前にそれぞれ u を隣接させた場合の調和平均と u 自身の出現確率の積を考慮することにより，出現率の高い u に対して前後ともに接続性が良い場合に強度が上がるように定義した．この項により，より形態素らしいものと接続する分割位置が高く評価されるため，不自然な分割位置を淘汰する効果が見込まれる．

5 複数の分割位置を選択する方法

前節による分割手法は分割する場合にどのような位置で分割するべきかを判断する基準であるが，分

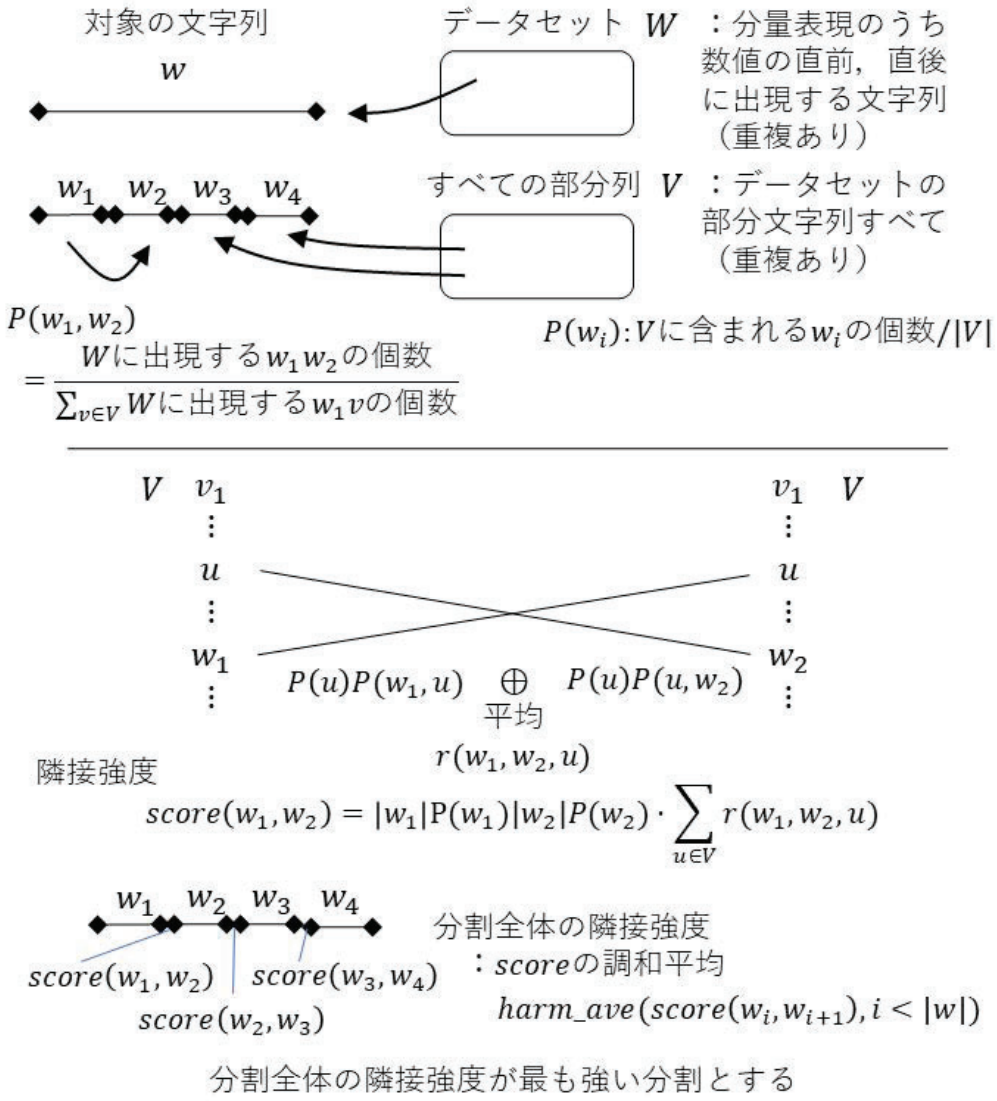

図 1　隣接強度の算出方法

割がない全体をひとつの形態素としたほうが良いかどうかの判定はできない．以下のアルゴリズムを用いて，対象記号列全体をひとつの形態素であるかどうかを判定する．

[単一形態素判定アルゴリズム]

1. データ w における判定対象文字列を $w \in W$ とする．
2. w のすべての部分文字列 v について，$v \in W$ でありかつ $P(v) > P(w)$ ならば，w は複数の形態素から成り立つものとして，分割を行う．
3. $P(v) \leq P(w)$ ならば，w 全体でひとつの形態素であると判断する．

複数の形態素からなる文字列の場合は，w に対する分割の中から，分割の隣接強度が最も高いものを求め，w を分割する．ここで分割の隣接強度は，それぞれの分割位置の隣接強度の調和平均によって求められる．

[分割の隣接強度算出アルゴリズム]

1. 判定対象文字列 w と算出対象の分割を $(v_0, v_1, \cdots, v_k)$ とする．
2. すべての $0 \leq i \leq k-1$ について，$s_i = score(v_i, v_{i+1})$ とする．
3. $s_i (0 \leq i \leq k-1)$ の調和平均を分割の隣接強度 S とする．

以上をまとめると，以下のアルゴリズムで $w \in W$ から助数詞および計量器の抽出が行える．

[全体アルゴリズム]

1. 単一形態素判定アルゴリズムを実行する．
2. w がひとつの形態素の場合：
   - w が数値の直前に出現するもの w_p ならば，計量器とする．
   - w が数値の直後に出現するもの w_s ならば，助数詞とする．
3. w が複数の形態素からなる場合：w に対するすべての分割について，その分割の隣接強度 S を求める．
4. 前ステップで最も高い隣接強度を持つ分割を $(v_0, v_1, \cdots, v_k)$ とする．
   - w が数値の直前に出現するもの w_p ならば，v_k を計量器とする．
   - w が数値の直後に出現するもの w_s ならば，v_0 を助数詞とする．

6 評価実験

本手法を用いて分量表現の前後に出現する語句の解析を行った．以下に実験に用いたデータの詳細を示す．

レシピ総数	1,715,595
食材名総数 = 分量表現総数	12,300,740
w_p の異なり数	7,388
w_p のすべての部分文字列の異なり数	85,966
w_s 異なり数	22,080
w_s のすべての部分文字列の異なり数	213,435

実験の結果，数値の前に隣接する語句は，分割を行わないと 7,000 種類ほどであるが，分割後はおよそ 5,000 種類と大きな減少が見られた．このことから，本手法による分割が適切な形態素を抽出していることがわかる．

次に数値の前に接続する文字列 w_p に関して，出現数が多い上位 100 種類の語句に対して，本手法により計量器が抽出できたかを確認する．

正しく計量器が抽出できた項目数	58/100
「大」「さじ」など細かくしすぎたもの	29/100
その他不正解	13/100

計量器をうまく抽出できた例として，「大さじ山盛り」を（大さじ，山盛り）と分解した．また，一般の形態素解析ではふたつに分割されることの多い「ティースプーン」などは単独で分割されない結果となった．分割に失敗する例としては「小さし」など誤記と思われる表現に対しては（小，さ，し）と分割する結果となった．「みじん切り小さじ」に対しては（みじん切り，小さじ）と分割されたが「刻んで大さじ」に対しては（刻んで，大，さじ）となり，修飾語によっても結果が変化することがあった．

同様に数値の直後に接続する文字列 w_s に関して，出現数が多い上位 100 この中で本手法により助数詞が抽出できたかを確認する．

正しく助数詞が抽出できた項目数	97/100
不正解	3/100

うまく抽出ができた例として，「本分」を（本，分）と分割し，「程度」を（程度）のまま分割されない結果となった．不正解は「つまみ」「かけら」「つかみ」であり，それぞれ（つ，ま，み）（かけ，ら）（つ，か，み）となった．

比較実験として，部分文字列の出現数のみに着目したスコア

$$score(v_i, v_j) = |v_i|P(v_i) \cdot |v_j|P(v_j)$$

を用いた分割抽出も行った．この実験でも前記と同様に出現数上位 100 個の w_p に対して性能を計測した．

正しく計量器が抽出できた項目数	44/100
「大」「さじ」など細かくしすぎたもの	18/100
その他不正解	38/100

この結果より，隣接文字列の出現数のみに着目すると性能が低下することがわかる．特に完全に形態素の境界でない部分を分割する不正解が増えていることから本研究による提案手法の有効性が示された．一方，数値の後に接続する w_s に対する性能は変化しなかった．

7 おわりに

本研究により文字列間の隣接強度を出現数から定義することにより，レシピの分量表現を形態素解析を行った．その結果，助数詞および計量器の抽出において文字列の出現数のみの場合より性能向上が確認できた．今後の課題として，材料表の食材名に応じて異なる助数詞の間の分量変換や数値の入っていない「少々」「適量」などの分量表現の理解が挙げられる．

謝辞

本研究では、国立情報学研究所の IDR データセット提供サービスによりクックパッド株式会社から提供を受けた「クックパッドデータセット」を利用した.

参考文献

[1] 山肩洋子, 今堀慎治, 森信介, 田中克己. 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. 信学論, Vol. J99-D, No. 4, pp. 378–391, 2016.

[2] H. Nanba, Y. Doi, T. Takezawa, K. Sumiya, and M. Tsujita. Construction of a cooking ontology from cooking recipes and patents. In **Proceedings of Workshop on Smart Technologyfor Cooking and Eating Activities : CEA2014 (CEA 2014)**, pp. 507–516, 2014.

[3] 森信介, 長尾眞. n グラム統計によるコーパスからの未知語抽出. 情報処理学会論文誌, Vol. 39, No. 7, pp. 2093–2100, 1998.

[4] 白井智, 鳥井修, 金井達徳. 反復文字列階層グラフによる文書からのキーワード自動抽出. 日本データベース学会 Letters, Vol. 4, No. 1, pp. 1–4, 2005.

[5] クックパッド株式会社 (2015). クックパッドデータ. 国立情報学研究所情報学研究データリポジトリ. (データセット), 2015.