法令文書を対象にした並列構造解析

松山　宏樹　　白井　清昭　　島津　明
北陸先端科学技術大学院大学 情報科学研究科
{hiroki-matsuyama, kshirai, shimazu}@jaist.ac.jp

1 はじめに

法律には特有の言い回しや表現方法があり，法令文を解析する際にはこれらに留意する必要がある．特に，法令文書では複雑な並列構造がよく使われていることから，法令文を解析するためには並列構造を正しく認識することが重要である．法令文における並列構造には以下のような特徴がある [6].

- 「及び」「並びに」は論理積を,「若しくは」「又は」は論理和を表わす.
- 階層構造を持つ並列構造がよく出現する．このとき,「並びに」は「及び」よりも,「又は」は「若しくは」よりも上位の並列構造を表わす.
- 3 つ以上の句の並列関係を表わす並列構造がよく出現する.

本研究は法令文を対象とした並列構造解析を目的とする.

並列構造解析に関する先行研究としては，特に長い文を対象に文節列の類似性を基に並列構造を検出する研究 [4]，並列構造解析と構文解析，格解析を統合した確率モデルを提案した研究 [3]，英語を対象に階層構造を持つ並列構造の解析モデルを提案した研究 [1] などがある．これらの研究に共通するのは，並列関係にある句が互いに類似しているという性質を手がかりとしている点である．本研究でも基本的には句の類似性を基に並列構造を解析するが，法令文の特徴を考慮し，法令ドメインに特化した解析アルゴリズムを提案する [5].

2 提案手法

2.1 概要

本論文では，並列構造は以下の要素から構成されると定義する.

$$\text{並列構造} \quad = \quad (pf_i \text{、})^* \;\; pf_1 \;\; key \;\; pb$$

key (並列キーと呼ぶ) は「又は」「と」のような並列構造を示唆する語を表わす．pf_i(前方句), pb(後方句) はそれぞれ並列関係にある句のうち並列キーの前方，後方に出現するものを指す．法令文では，3 つ以上の句が並列関係にある場合は，2 つ以上の前方句を読点で並べた後，並列キー，後方句が続く．並列キーに一番近い前方句を pf_1，以下順に pf_2, pf_3 と表記する．例文 ⟨1⟩ は法令文書における並列構造の例である.

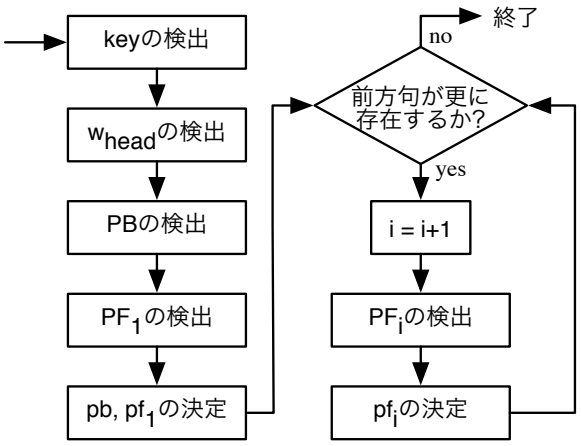

図 1: 並列構造解析の処理の流れ

及び, 若しくは	>	並びに, 又は	>	と, や, かつ, その他
(優先順位 1)		(優先順位 2)		(優先順位 3)

図 2: 並列キーの一覧

⟨1⟩ 四百八十から保険料納付済期間の月数$_{(pf_3)}$、保険料四分の一免除期間の月数$_{(pf_2)}$、保険料半額免除期間の月数$_{(pf_1)}$ 及び$_{(key)}$ 保険料四分の三免除期間の月数$_{(pb)}$ を合算した月数を控除して得た月数を限度とする

提案する並列構造解析アルゴリズムの処理の流れを図 1 に示す．前処理として，解析対象の文を JUMAN[1] を用いて形態素解析する．まず最初に並列キー key を検出する．本研究で取り扱う並列キーの一覧を図 2 に示す[2]．次に前方句の主辞 w_{head} を検出する．ここでは，w_{head} は key の直前に出現する語 (直前の語が読点の場合はその前の語) とする．さらに，後方句の候補の集合 PB，前方句の候補の集合 PF_1 をそれぞれ検出する (詳細は 2.2, 2.3 項で述べる)．これらの集合から後方句 pb, 前方句 pf_1 を決定する (2.4 項)．pf_1 の前に更に別の前方句 $pf_i (i \geq 2)$ が存在するときは，それらを順に検出する (2.5 項)．最後に並列構造 $(pf_n, \cdots, pf_1, key, pb)$ を出力する.

[1] http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN
[2] 優先順位については 3 節で述べる.

2.2 後方句の候補の検出

key と w_{head} を検出した後，後方句の候補を検出する．具体的には，候補となる句の始点と終点を決定する．後方句の始点は常に key の直後の語 (読点のときはさらにその次の語) とする．一方，後方句の終点は，以下に該当する語を key より後方へ探索し，その全てを候補とする．ただし，読点，他の並列キーもしくは句点が現われた時点で探索を終了する．

- w_{head} の品詞が名詞のとき
  文節内の最後に出現する自立語であり，かつ品詞が名詞である語．さらに，以下のように後方句の終点の候補を絞り込む．
  - w_{head} と全く同じ単語があれば，それらのみを後方句の終点の候補とする．
  - 得られた候補のうち，key に一番近いもの，および w_{head} との意味的類似度が大きい上位 3 つの語のみを後方句の終点の候補とする．単語間の意味的類似度 sim_w は日本語語彙大系 [2] を用いて式 (1) で求める[3]．

$$sim_w(w_i, w_j) = \frac{2 \times d_c}{d_i + d_j} \tag{1}$$

- w_{head} の品詞が動詞のとき
  文節内の最後に出現する自立語であり，かつ品詞が動詞である語．
- w_{head} の品詞が助詞のとき
  文節内の最後に出現し，かつ品詞が助詞である語．

得られた後方句の候補を $PB = \{pb_j\}$ とする．

2.3 前方句の候補の検出

前方句の候補となる句の始点と終点を決定する．終点は常に w_{head} とする．w_{head} より前方を探索し，文節の先頭にある全ての語を始点の候補とする．ただし，読点，他の並列キーもしくは文頭に到達した時点で探索を終了する．得られた前方句の候補を $PF_1 = \{pf_{1k}\}$ とする．

2.4 句の類似度の計算

後方句の候補 PB と前方句の候補 PF_1 から後方句と前方句を 1 つずつ選択する．ここでは並列関係にある句は互いに似ていると仮定し，類似度の高い句の組を選択する (式 (2))．

$$(pb, pf_1) = \underset{pb_j \in PB,\ pf_{1k} \in PF_1}{\arg\max} sim_p(pb_j, pf_{1k}) \tag{2}$$

[3] d_i, d_j, d_c はそれぞれ日本語語彙大系における w_i, w_j, w_i と w_j の共通上位ノードの深さ．

$sim_p(a, b)$ は句の類似度で，単語単位のアライメントを基に算出する．$a = wa_1 \cdots wa_n$, $b = wb_1 \cdots wb_m$ とし (wa_i, wb_j はそれぞれ句 a, b を構成する単語)，句 a と b のアライメント $ALIGN$ を次のように定義する．

$$ALIGN = \{a_k\}, \quad \text{但し}, a_k = (wa_i, wb_j) \text{ or } (wa_i, \phi) \text{ or } (\phi, wb_j) \tag{3}$$

(wa_i, wb_j) は wa_i と wb_j に対応関係があることを，(wa_i, ϕ) 及び (ϕ, wb_j) はそれぞれ wa_i, wb_j に対応する語がないことを表わす．句の類似度は，可能なアライメントのうち，最もスコアが高いものの値とする．

$$sim_p(a, b) = \max_{ALIGN}\ score_A(ALIGN) \tag{4}$$

アライメントのスコア $score_A(ALIGN)$ は以下のように定義する．

$$score_A(ALIGN) = \frac{1}{|ALIGN|} \sum_{a_k} score_a(a_k) \tag{5}$$

$$score_a(a_k) = \gamma \cdot s\text{-}word(a_k) + (1 - \gamma) \cdot s\text{-}skip(a_k) \tag{6}$$

式 (5) に示したように，ALIGN のスコアは個々の対応関係 a_k のスコア $score_a(a_k)$ の平均値とする．$score_a(a_k)$ は，$s\text{-}word(a_k)$ と $s\text{-}skip(a_k)$ の重み付き和とする．$s\text{-}word(a_k)$ は単語間の類似度に応じて与えるスコアであり，$s\text{-}skip(a_k)$ は対応関係がない場合に対して与えられるスコア (ペナルティ) である．ともに 0 から 1 までの間の値をとる．重み γ は予備実験により 0.6 とした．

$a_k = (wa_i, wb_j)$ のとき ($wa_i \neq \phi$, $wb_j \neq \phi$)，$s\text{-}word(a_k)$ と $s\text{-}skip(a_k)$ を以下のように定義する．

$$s\text{-}word(a_k) = \begin{cases} 1 & \text{if } wa_i = wb_j \\ 0.9 & \text{if } wa_i \text{と } wb_j \text{がともに数字} \\ sim_w(wa_i, wb_j) \times 0.6 + 0.2 & \text{if [A]} \\ 0.1 & \text{if } wa_i \text{と } wb_j \text{の品詞が同じ} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$s\text{-}skip(a_k) = 1 \tag{8}$$

式 (7) における [A] の行は，wa_i と wb_j がともに日本語語彙大系に登録されている場合，式 (1) で定義される類似度 $sim_w(wa_i, wb_j)$ をスコアとすることを表わす．但し，0.2 から 0.8 までの間の値を取るようにスケーリングされている．一方，$s\text{-}skip(a_k)$ は最大値の 1 とする．

$a_k = (wa_i, \phi)$ もしくは (ϕ, wb_j) のとき，$s\text{-}word(a_k)$ と $s\text{-}skip(a_k)$ を以下のように定義する．

$$s\text{-}word(a_k) = 0 \tag{9}$$

$$s\text{-}skip(a_k) = \begin{cases} (1 - i/n) & \text{if } (wa_i, \phi) \text{ のとき} \\ (1 - j/m) & \text{if } (\phi, wb_j) \text{ のとき} \end{cases} \tag{10}$$

句 a :	保険	料	ϕ	納付	済	期間	の	月	数
句 b :	保険	料	半額	免除	ϕ	期間	の	月	数
	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
$s\text{-}word(a_k)$	1	1	0	0.52	0	1	1	1	1
$s\text{-}skip(a_k)$	1	1	0.625	1	0.5	1	1	1	1
$score_a(a_k)$	1	1	0.25	0.712	0.2	1	1	1	1

$$score_A = \frac{(1+1+0.25+0.712+0.2+1+1+1+1)}{9} = 0.796$$

図 3: アライメントに基づく句の類似度の計算例

式 (10) において，n, m は句 a, b の単語数である．すなわち，$s\text{-}skip(a_k)$ は，wa_i または wb_j の句の中での位置が末尾に近いほど低いスコアを与える．これは，日本語の主辞は句の最後に位置することから，主辞やそれに近い単語で対応関係がない場合に高いペナルティを与えるためである．一方，$s\text{-}word(a_k)$ は最低値の 0 とする．

さらに，法令文の特徴を考慮し，以下のヒューリスティクスを導入する．

- 「第」「条」「項」「号」は，式 (7) の計算の際，同じ単語に対応付けられているときのみスコアを 1 とし，それ以外は 0 とする．これらの単語は法令文の条文番号を表わす単語であり，他の単語と対応付けるのは不自然である．
- 「同」「前」「次」という単語が「条」「項」「号」の直前に出現するときは,「第」「条」「項」「号」および数字の連続で構成される複数の単語と対応付け，それに対して最大のスコアを与える．例えば『第七条第一項第二号』と『同項第三号』という 2 つの句に対しては,「第七条第一」と「同」を対応付け，そのときのスコアを 5 とする[4]．

『保険料納付済期間の月数』と『保険料半額免除期間の月数』という 2 つの句に対するアライメントとそのスコアの計算例を図 3 に示す．この 2 つの句については他にも可能なアライメントがあるが，式 (4) に示したように，スコアが最大となるアライメントを選択し，そのスコアを句の類似度とする．これは DP マッチング法により効率良く計算することができる．

2.5 2 つ目以降の前方句の検出

後方句 pb と最初の前方句 pf_1 を検出した後，2 番目以降の前方句を検出する．今，後方句 pb と $i-1$ 個の前方句 pf_l $(1 \leq l \leq i-1)$ が検出されているとき，i 番目の前方句 pf_i の検出を試みる．以下の条件を満たすときは pf_i が存在するとみなして前方句の探索を継続し，満たされないときは並列構造解析を終了する．

- 既に検出された前方句 pf_{i-1} の直前が読点である．
- その読点の直前の語 w_x と w_{head} の品詞が同じ．
- (w_{head} の品詞が名詞のとき) w_{head} と w_x の意味的類似度 (式 (1)) が 0.4 より大きい．

次に，前方句の候補 $PF_i = \{ pf_{ik} \}$ を作成する．pf_{ik} の終点は常に w_x とし，始点は 2.3 項で述べた手法で決める．この中から，既に検出されている前方句 pf_l 及び後方句 pb との類似度の和が一番大きいものを pf_i とする (式 (11))．

$$pf_i = \mathbf{arg\ max}_{pf_{ik} \in PF_i} \sum_{l=1}^{i-1} sim_p(pf_{ik}, pf_l) + sim_p(pf_{ik}, pb) \quad (11)$$

2.6 解析例

例文 ⟨1⟩ に対して並列構造を検出する処理の流れを説明する．まず，並列キー key として「及び」を検出し，次に前方句の主辞 w_{head} を「数」とする．後方句の候補 PB として { 保険料四分の三免除期間の月数, 保険料四分の三免除期間の月数を合算した月数, 保険料四分の三免除期間の月数を合算した月数を控除して得た月数 }, 前方句の候補 PF_1 として { 月数, 保険料半額免除期間の月数 } を得る．全ての句の組み合わせの中から式 (4) の類似度が最大となるものを選択し，pb=(保険料四分の三免除期間の月数)，pf_1=(保険料半額免除期間の月数) とする．さらに前方句の検出を試み，pf_2=(保険料四分の一免除期間の月数)，pf_3=(保険料納付済期間の月数) とする．pf_3 の前には読点はないので，解析を終了する．

3 階層的並列構造の検出

本節では階層的な並列構造を検出するためのアルゴリズムについて述べる．本研究では，下位の並列構造から上位の並列構造という順序で並列構造を逐次検出する．まず，1 節で述べた法令文における特徴を考慮し，並列キーに図 2 に示した優先順位をつける．優先順位の高い並列キーから，同じ優先順位を持つ並列キーが複数ある場合は先頭に出現するものから順に，それを含む並列構造を検出する．

さらに，2 節で述べた解析アルゴリズムを以下のように変更する．

- 2.2 項，2.3 項で後方句もしくは前方句の候補を検出する際，既に検出した並列構造の内部は後方句，前方句の境界としない．

[4](第, 同), (七, 同), (条, 同), (第, 同), (一, 同) の 5 つの対応関係があるとみなし，それぞれ最大のスコア 1 を持つとみなすため．

- 同様に後方句もしくは前方句の候補を検出する際，既に検出した並列構造の境界は必ず後方句，前方句の候補の境界とする．
- 2.4 項で句の類似度を計算する際，句の内部に既に検出されている並列構造がある場合は，それを後方句のみに置き換えて式 (4) の句の類似度を計算する．これは，前方句と後方句のどちらか一方のみに下位の並列構造がある場合，両者の類似度が低く見積られることを避けるためである．

3.1 解析例

⟨2⟩ 国民年金手帳の様式$_{(pf_1^1)}$ 及び$_{(key^1)}$ 交付$_{(pb^1)}$ $\Big)_{(pf_1^2)}$
その他$_{(key^2)}$ 国民年金手帳に関して必要な事項$_{(pb^2)}$
は、厚生労働省令で定める。

例文 ⟨2⟩ では 2 つの並列キーがあり,「及び」「その他」の順に並列構造を検出する．まず,「及び」を含む並列構造 (pf_1^1, key^1, pb^1)『様式及び交付』を検出する．次に,「その他」を並列キーとする並列構造の検出を試みる．前方句の候補として『国民年金手帳の様式及び交付』が，後方句の候補として『国民年金手帳に関して必要な事項』があるとき，前者は既に検出した下位の並列構造を含むため，これを後方句 pb^1 に置き換え,『国民年金手帳の交付』と『国民年金手帳に関して必要な事項』の類似度を計算する．最終的に (pf_1^2, key^2, pb^2) を上位の並列構造として検出する．

4 評価実験

国民年金法の法令文を対象に提案手法の評価実験を行った．同法の先頭から 300 個の法令文に対して正解とする並列構造を人手で付与した．このうち 1〜200 番目の文は開発データとして並列構造解析アルゴリズムの設計や改良のために使用した．一方，201〜300 番目の文は評価データとして使用した．開発データ，評価データにおける並列構造の数は，階層的な並列構造を構成するものも含めて，それぞれ 188，68 である．本実験では提案手法並びに KNP ver. 3.01[5]で並列構造解析を行い，両者を比較した．但し，KNP の出力において，本研究で使用している並列キーを含まない並列構造は評価の対象から除外した．

実験結果を表 1 に示す．表 1 では，検出された並列構造ならびに前方句・後方句の精度 (P)，再現率 (R)，F 値 (F) を示している．

提案手法は開発データ，評価データともに KNP を上回っている．精度よりも再現率の差が大きいが，KNP で

[5] http://nlp.ist.i.kyoto-u.ac.jp/index.php?KNP

表 1: 実験結果

		並列構造			前方句, 後方句		
		P	R	F	P	R	F
提案	開発	0.61	0.62	0.62	0.78	0.77	0.78
	評価	0.47	0.54	0.50	0.61	0.68	0.65
KNP	開発	0.49	0.36	0.41	0.76	0.46	0.58
	評価	0.27	0.26	0.26	0.50	0.44	0.47

は階層的な並列構造の検出にほとんど失敗していることがその要因となっている．しかしながら，開発データに対してさえ並列構造全体の F 値が 0.62 であることから，提案手法の解析精度は十分に高いとは言えない．解析誤りの主な原因を以下に述べる．まず，前方句と後方句の長さが大きく異なるときに並列構造の検出に失敗する場合が多かった．現在は前方句と後方句の類似性を主な手がかりとしているが，並列構造と他の要素との係り受け関係も考慮しないと，前方句と後方句の長さが異なる場合に対応できない．また，前方句は並列キーに近いものから順に検索するが，途中の前方句の検出に失敗し，その時点で更に前方にある前方句の探索を中止している場合も多かった．評価データの精度が開発データと比べて低下している要因は，格助詞の「と」を誤って並列キーと検出する場合が多いことであった．

5 おわりに

本論文では，法令文を対象とし，3 つ以上の句が並ぶ並列構造や階層的な並列構造の解析手法について述べた．今後は，4 節で述べた解析誤りの分析結果を基に提案手法を改良し，解析精度を更に向上させたい．

参考文献

[1] Kazuo Hara, Masashi Shimbo, Hideharu Okuma, and Yuji Matsumoto. Coordinate structure analysis with global structural constraints and alignment-based local features. In *Proceedings of the 47th ACL and the 4th IJCNLP*, pp. 967–975, 2009.

[2] 池原悟, 宮崎正弘, 白井諭, 横尾昭男, 中岩浩己, 小倉健太郎, 大山芳史, 林良彦. 日本語語彙大系 — 全 5 巻 —. 岩波書店, 1997.

[3] Daisuke Kawahara and Sadao Kurohashi. Probabilistic coordination disambiguation in a fully-lexicalized Japanese parser. In *Proceedings of the Joint Conference of EMNLP and CNLL*, pp. 306–314, 2007.

[4] 黒橋禎夫, 長尾眞. 並列構造の検出に基づく長い日本語文の構文解析. 自然言語処理, Vol. 1, No. 1, pp. 35–58, 1994.

[5] 松山宏樹. 法令文書を対象とした並列構造解析の精緻化. 修士論文, 北陸先端科学技術大学院大学, 3 2012.

[6] 上田章, 笠井真一. 条例規則の読み方・つくり方 – 市町村の実例を中心として. 学陽書房, 2000.