日本語法令文における定型表現の半自動獲得

Semi-automatic Pattern Acquisition from Japanese Legal Sentences

苅谷 恵介 *，小川 泰弘，外山 勝彦 (名古屋大学)
Keisuke Kariya, Yasuhiro Ogawa, and Katsuhiko Toyama (Nagoya University)

1　はじめに

近年、国際取引の円滑化や、対日投資の促進、海外での法整備支援の観点から、我が国の法令の翻訳が必要とされている [1, 2]。しかし、法令の翻訳は、これまで所管府省や民間によって個別に行われてきた。そのため、同じ用語や言い回しなどの訳語が必ずしも統一されておらず、利用者に誤解を生じさせる場合もあった。

それに対して、日本政府は「法令用語日英標準対訳辞書」[1](標準対訳辞書) を作成することにより、翻訳ルールを策定している。現在、この翻訳ルールに準拠して、法令の翻訳が進められている。

しかし、法令文では、文の意味を一定に保ち、解釈に誤りを生じさせないようにするために、図 1 に示すような定型表現が多用される [3]。法令の翻訳においては、単語だけでなく、それら定型表現の翻訳も統一される必要がある。しかし、標準対訳辞書に収録されている定型表現は 89 個に過ぎず、実際、図 1 に示す定型表現はいずれも収録されていない。したがって、定型表現を網羅した対訳辞書の作成が望まれる。

そこで、本研究では、対訳辞書に収録する定型表現を獲得することを目的とする。そのために、系列パターンマイニングを用いる。しかし、法令文に対して系列パターンマイニングを適用すると、低頻度の定型表現が獲得できない。そこで、法令文の述語に着目して、法令文を内容ごとに分類する手法を提案する。また、表現の揺らぎに由来する類似表現が大量に抽出されるという問題が生じる。それに対して、法令文の集合の類似度を利用して不要な類似表現を削除する手法を提案する。

2　系列パターンマイニング

標準対訳辞書や解説書に収録されている定型表現は、様々な長さのものがあり、その多くは不連続な単語の列となっている。そこで、任意の長さの必ずしも連続でない単語の列を候補表現とし、法令文を系列とみなして系列パターンマイニングの手法を適用する。

[1] http://www.cas.go.jp/jp/seisaku/hourei/data1.html

> …者は、…年以下の懲役又は…万円以下の罰金に処する。
> …法は、廃止する。
> …については、…の規定は、適用しない。
> …には、…税を課さない。
> この法律の施行前にした行為に対する罰則の適用については、なお従前の例による。

図 1: 法令文における定型表現の例

本節では、定型表現の獲得のために用いる系列パターンマイニングについてまとめる。

系列パターンマイニング [4] は、系列の集合と最小サポートとよばれる出現頻度の閾値 θ が与えられたとき、θ 回以上出現するすべての部分系列を列挙する問題であり、次のように定式化できる。

$I = \{i_1, i_2, \ldots, i_m\}$ をアイテム集合とする。系列をアイテムの順序リストとし、系列全体の集合を I^* とする。系列 $s = a_1 a_2 \ldots a_n \in I^*$ に対して、s の長さを $|s|$ で表す。また、2 つの系列 $s_a = a_1 a_2 \ldots a_m$, $s_b = b_1 b_2 \ldots b_n$ に対して、$i_1, i_2, \ldots, i_m (1 \leq i_1 < i_2 < \cdots < i_m \leq n)$ が存在して、$a_k = b_{i_k} (1 \leq k \leq m)$ のとき、s_a は s_b の部分系列である、または、s_a は s_b に出現するといい、$s_a \sqsubseteq s_b$ と表す。特に、$s_a \neq s_b$ のとき、$s_a \sqsubset s_b$ と表す。

系列データベース D を組 (sid, s) の集合とする。ここで、sid は系列 s に対応する識別子である。系列データベース D において系列 s_p が出現する系列の集合を $occ_D(s_p) = \{(sid, s) \in D | s_p \sqsubseteq s\}$ とし、D における s_p のサポートを $sup_D(s_p) = |occ_D(s_p)|$ とする。

系列パターンマイニングは、系列データベース D と最小サポート θ が与えられたとき、次の頻出系列の集合 $FS_D(\theta)$ を求める問題である。

$$FS_D(\theta) = \{s_p \in I^* | sup_D(s_p) \geq \theta\}$$

本研究では、系列パターンマイニングアルゴリズム BIDE[5] を用いて $FS_D(\theta)$ を求める。

3 提案手法

従来手法を法令文における定型表現の獲得に利用すると、以下の２つの問題が生じる。

1. 低頻度の定型表現が抽出できない。
2. 類似した表現ばかりが抽出されてしまう。

本節では、これらの問題を解決する手法を提案する。

3.1 系列データベースの分割

法令文の書き方 (法制執務) に関する解説書 [3] に収録されている定型表現 132 種は、平成元年〜18 年に新規に制定された法律 2,447 本[2]から抽出した法令文 184,614 文において、平均 272 文出現した。このような低頻度の定型表現を抽出するためには、非常に低い最小サポートを与える必要がある。しかし、最小サポートを低くすると、抽出される単語の組み合わせが爆発的に増加し、膨大な計算時間を必要とするという問題がある。

それに対して、法令文を内容ごとに分類することにより対処する。解説書に収録されている定型表現は、語彙の定義をする表現、刑罰を定める表現、施行日を定める表現など、定型表現のおおまかな内容ごとに分類されている。各分類では、内容の細かな違いにより複数の定型表現を持つ場合があるが、語彙の定義をする表現であれば「をいう」、刑罰を定める表現であれば「に処する」、施行日を定める表現であれば「から施行する」のように文末の述語を中心とした表現が共通して使われている。このような表現は、次の 3 つの要素から構成されている。

格助詞
　述語に先行して、述語の意味を限定

述語
　文の内容を規定

モダリティ表現
　述語に後続して義務や禁止などの意味を付加

そこで、文末の「格助詞＋述語＋モダリティ表現」を述語表現とし、法令文の集合を共通の述語表現を持つ集合に分割し、それぞれを系列データベースとする。個々の系列データベースは、同じ内容、すなわち同じ定型表現を持つ文の集合となるため、系列データベースにおける定型表現の出現頻度は相対的に増加する。

[2] http://www.shugiin.go.jp/index.nsf/html/index_housei.htm

表 1: 類似した候補表現の例

id	候補表現	頻度
s_1	この/法律/は/、/公布/の/日/から/施行/する/。	100
s_2	**この/法律/は/公布/の/日/から/施行/する/。**	101
s_3	の/規定/は/、/について/は/、/適用/し/ない/。	100
s_4	**の/規定/は/、/適用/し/ない/。**	101
s_5	**について/は/、/適用/し/ない/。**	102

3.2 不要表現の削除

抽出した候補表現を順位付けすると、上位に類似した候補表現が集中して出現するという問題がある。

ここで、類似した候補表現の例を表 1 に示す。s_1 と s_2 に注目すると、s_2 は s_1 の部分系列であり、読点が省略された候補表現となっている。また、s_1 は 100 回、s_2 は 101 回出現しているが、s_2 の出現頻度 101 回のうち 100 回は s_1 の部分系列として出現したものであり、s_2 そのものが出現したのはわずか 1 回である。そこで、s_2 は s_1 に対して語の省略という表現の揺れがあったために抽出されたもの、すなわち、不要な部分系列表現 (不要表現) とみなし、候補表現のリストから削除する。

同様に、表 1 の s_3 に対して、s_4 および s_5 は部分系列であり、頻度の差がわずかであるため、s_4 および s_5 は不要表現として削除する。しかし、最小サポートを $\theta = 101$ とした場合、s_4 および s_5 は不要表現であるにもかかわらず、s_3 が候補表現として出力されないため削除することができない。

ここで、s_3 と s_4 および s_5 との関係を考える。s_4 および s_5 はそれぞれ s_3 の部分系列だから、$occ_D(s_3) \subseteq occ_D(s_4)$ かつ $occ_D(s_3) \subseteq occ_D(s_5)$、すなわち、$occ_D(s_3) \subseteq occ_D(s_4) \cap occ_D(s_5)$ である。したがって、s_3 の頻度は最大でも $|occ_D(s_4) \cap occ_D(s_5)|$ となる。そこで、s_3 の頻度を $|occ_D(s_4) \cap occ_D(s_5)|$ とみなして、s_4 および s_5 が不要表現であるかどうかを判定する。

3.2.1 削除手法

部分系列関係にある 2 つの候補表現 $s_a, s_b (s_a \sqsubset s_b)$ に対して、s_b の部分系列 s_a が不要表現であるかどうかを判定する必要がある。

表 1 の例では、頻度の差はわずか 1 であったが、出現頻度の高い表現に対して表現の揺れがあった場合、

頻度の差も高くなると考えられる。そこで、2 つの系列の出現集合の類似度を求め、類似度の閾値 δ を与えて不要表現かどうかを判定する。集合 A, B 間の類似度は、次のコサイン類似度で与える。

$$sim(A,B) = \frac{|A \cap B|}{\sqrt{|A| \times |B|}}$$

類似度の閾値 δ が与えられたとき、任意の系列 $s_a, s_b \in FS_D(\theta)$ に対して、$sim(occ_D(s_a), occ_D(s_b)) \geq \delta$ のとき、次のように不要表現を削除する。

1. $s_a \sqsubset s_b$ ならば、s_a を削除する。
2. そうでないとき、s_a および s_b を削除する。

4 評価実験

4.1 実験対象

平成元年〜18 年に新規に制定された法律 2,447 本から抽出した法令文 184,614 文を用いて実験を行った。法令文を ChaSen によって形態素解析し、単語の出現形の列からなる系列データベースを作成する。この系列データベースを述語表現 6,720 個によって分割し、各要素数が 10 以上の系列データベース 1,171 個 (延べ 170,814 文) を対象として、定型表現の獲得を行う。

4.2 実験条件

系列データベース D に対して、系列パターンマイニングにおける最小サポートを $\theta = 0.1 \times |D|$ として候補表現を抽出する。また、不要表現の削除における類似度の閾値を $\delta = 0.95$ とした。

4.3 候補の評価

各系列に対する重要度の指標として C-value[6] を用いる。C-value は単語 n-gram に対する指標のため、本研究では、系列 s_p に対する C-value $CV(s_p)$ を次のように定義する。

$$CV(s_p) = \begin{cases} |s_p| \times n(s_p) & , \exists s \ s_p \sqsubset s \\ |s_p| \times \left(n(s_p) - \frac{t(s_p)}{c(s_p)} \right) & , otherwise \end{cases}$$

ここで、

$$n(s_p) = sup_D(s_p),$$
$$T_{s_p} = \{ s_q \in FS_D(\theta) | s_p \sqsubset s_q \},$$
$$t(s_p) = \sum_{s_q \in T_{s_p}} sup_D(s_q),$$
$$c(s_p) = |T_{s_p}|.$$

4.4 評価指標

評価指標として再現率と被覆率を用いる。再現率は、既知の定型表現に対してどれだけ獲得することができたかを示す。被覆率は、獲得した定型表現のうち少なくとも 1 つを含む文の割合を表す。

4.4.1 再現率

正解とする定型表現の集合 C と獲得した定型表現の集合 P が与えられたとき、P の再現率 $Recall(P)$ を次のように定義する。

$$Recall(C,P) = \frac{|\{ s_c \in C | \exists s_p \in P \ s_c = s_p \}|}{|C|}$$

しかし、法令文における定型表現は、解説書によって差異があることが多い。そこで、正解として、完全一致の他に、編集距離を用いた系列間の類似度尺度 $sim(s_c, s_p)$ と類似度の閾値 σ を用いて、$sim(s_c, s_p) \geq \sigma$ の場合を部分一致とし、次の再現率も考える。

$$Recall(C,P) = \frac{\left| \left\{ s_c \in C | \max_{s_p \in P} sim(s_c, s_p) \geq \sigma \right\} \right|}{|C|}$$
$$sim(a,b) = 1.0 - \frac{EditDistance(a,b)}{\max(|a|,|b|)}$$
$$\sigma \in [0,1]$$

4.4.2 被覆率

系列データベース D と定型表現の集合 P が与えられたとき、D において P の表現を 1 つ以上含む系列の割合を被覆率 $Coverage(D,P)$ とする。すなわち、

$$Coverage(D,P) = \frac{|\{ (sid,s) \in D | \exists s_p \in P \ s_p \sqsubseteq s \}|}{|D|}.$$

5 結果と考察

標準対訳辞書および解説書に収録されている定型表現のうち、述語表現を含み、10 文以上に出現したものを正解集合とした。獲得した定型表現の C-value の上位 10 件までの再現率および被覆率を表 2 に示す。

完全一致に対する再現率は非常に低い値となっているが、同じ内容を表す定型表現であっても、収録されている解説書によって形式が異なることが多いためであ

表 2: 上位 10 件までの再現率と被覆率

上位 n 件		1	2	3	4	5	6	7	8	9	10
標準対訳辞書に対する再現率 (%)	完全一致 ($\sigma = 1.00$)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	部分一致 ($\sigma = 0.75$)	27.8	33.3	33.3	33.3	33.3	33.3	33.3	33.3	33.3	33.3
	部分一致 ($\sigma = 0.50$)	55.6	55.6	55.6	55.6	55.6	55.6	55.6	55.6	66.7	66.7
解説書に対する再現率 (%)	完全一致 ($\sigma = 1.00$)	3.2	4.8	6.5	9.7	9.7	9.7	11.3	11.3	11.3	12.9
	部分一致 ($\sigma = 0.75$)	25.8	30.6	33.9	37.1	37.1	37.1	37.1	37.1	38.7	42.0
	部分一致 ($\sigma = 0.50$)	56.5	59.7	66.1	66.1	67.7	71.0	74.2	74.2	79.0	80.6
被覆率 (%)		81.9	88.7	91.6	93.1	94.0	94.4	94.6	94.9	95.1	95.5
定型表現の数		1,153	2,300	3,436	4,567	5,690	6,799	7,900	8,987	10,066	11,135

表 3: 標準対訳辞書および解説書に収録されていない定型表現（一部）

id	定型表現	頻度
1	は/、/が/とき/は/、/に対し/、/を/命ずる/こと/が/できる/。	986
2	の/登記/の/申請/書/に/は/、/を/添付/し/なけれ/ば/なら/ない/。	378
3	この/法律/に/規定/する/大臣/の/権限/は/、/で/定める/ところ/に/より/、/地方/に/委任/する/こと/が/できる/。	152
4	は/、/の/許可/を/受け/なけれ/ば/、/の/全部/又は/一部/を/休止/し/、/又は/廃止/し/て/は/なら/ない/。	61
5	の/場合/において/、/国会/の/閉会/又は/衆議院/の/解散/の/ため/に/両/議院/の/同意/を/得る/こと/が/でき/ない/とき/は/、/内閣/は/、/の/規定/に/かかわら/ず/、/同/項/に/定める/資格/を/有する/者/の/うち/から/、/委員/を/任命/する/こと/が/できる/。	16

る。また、部分一致 ($\sigma = 0.75$) のとき、上位 5 件までの結果をみると、標準対訳辞書および解説書に収録されている定型表現のうち、それぞれ 33.3%, 37.1%の既知の定型表現を獲得できたが、これでは十分とは言えない。獲得できなかった定型表現は、出現頻度が低かったか、その断片しか獲得できなかったと考えられる。

一方、被覆率については、上位 1 件 (延べ 1,151 個) の定型表現に対して 81.9%と、高い値を示した。すなわち、獲得した定型表現をすべて対訳辞書に収録した場合、約 8 割の法令文に対して、定型表現の統一的な翻訳の支援ができる可能性がある。

また、標準対訳辞書および解説書に収録されていない定型表現として表 3 を獲得することができた。

6 おわりに

法令文における定型表現を獲得するために、既存の法令文を対象とした系列パターンマイニングの手法による候補表現の抽出、用語抽出の順位付け指標を用いた候補表現の順位付けを行った。その際、低頻度の定型表現の獲得における計算量の問題に対しては、述語表現に着目した法令文の分類方法を、また、順位付け指標で区別できない不要な表現による性能の低下に対しては、出現集合の類似度を用いた削除方法を提案した。

今後の課題として、本研究により獲得した定型表現に対して対訳辞書に収録すべきかを人手により判定すること、述語表現の定義の見直し、不要表現の削除における類似度指標の検討が挙げられる。

参考文献

[1] 外山, 小川, 松浦: 日本法令翻訳システムの構想, ジュリスト, No. 1281, pp.2-5, 2004.

[2] 法令外国語訳・実施推進検討会議: 最終報告, `http://www.cas.go.jp/jp/seisaku/hourei/houkoku.pdf`, 2006.

[3] 大島: 法令起案マニュアル, ぎょうせい, 2004.

[4] Agrawal, R. and Srikant, R.: Mining sequential patterns, *Proceedings of the 11th International Conference on Data Engineering*, pp.3-14, 1995.

[5] Wang, J. and Han, J.: BIDE: Efficient Mining of Frequent Closed Sequences, *Proceedings of the 20th International Conference on Data Engineering*, pp.79-90, 2004.

[6] Frantzi, K.T. and Ananiadou, S.: Extracting nested collocations, *Proceedings of the 16th International Conference on Computational Linguistics*, pp.41-46, 1996.