極小言語戦略による文テンプレート獲得

服部 一浩 [*1]　横野 光 [*2]　相澤 彰子 [*2*1]
[*1]東京大学大学院　[*2]国立情報学研究所
{khattori, yokono, aizawa}@nii.ac.jp

1 はじめに

対話システムの応答文生成といった文生成のタスクにおいて文テンプレートを用いた研究は古く, 文法的に正しく自然な文を生成しやすいため実際のアプリケーションに用いられる例は多い. 文テンプレートとは空欄を含んだ文であり, 空欄に一語以上の語の列を埋めることで文が生成される. また, 書き言葉であるか話し言葉であるか, どんな分野の文章であるかといったドメインによって使われやすい常套句や構文があり, テンプレートに予めそれらを持たせることで, そのドメインに自然な文を生成することが容易になる. このようなテンプレートは作文支援にも使うことができると期待できる. すなわち, 書こうとしている文章のドメイン (e.g. メール, 履歴書) に相応しいテンプレートの候補を示し, 人間は一つ選んだテンプレートが持つ空欄に語句を埋めることで適切な文が作成できる. 第二言語学習者にとっては, そのドメインにおける自然な言い回しが学習できる.

人手によるテンプレートの作成はコストが掛かるため, 自動的な獲得手法が必要となる. Filatova らはドメインに特化した文生成のためのテンプレート獲得手法として, コーパス中の文を統語解析器によって構文木に変換し, 頻出する部分木パターンを発見し, テンプレートを生成する手法を提案した [1]. この方法には, 統語解析器の精度に依存する問題と, 木構造を直接扱うことによる計算量的コストがある問題とがある. Pei らによる PrefixSpan [2] は, 文を単語列として扱い, 部分列として頻出するパターンを効率よく列挙する手法であり, Filatova らによる手法と同様にして PrefixSpan を用いてテンプレートを生成することができる. しかしながら, 頻度ベースでパターンを発見することによる方法には問題がある. コーパスが実際にあるテンプレートの集合によって生成されたと仮定すると, あるテンプレートはコーパス中に頻出することが期待でき, 頻出するパターンの発見によって確かにテンプレートが獲得できるが, 必ずしも頻出するパターンが良いテンプレートであるわけではない. 或いは, 比較的頻出しないパターンが悪いテンプレートであるわけではない. 例えば挨拶のフレーズ (e.g. “Hi.”) と冠詞だけを持ったテンプレート (e.g. “The □”) とでは後者の方が頻出するとしても, テンプレートとしては前者は使われ方が人間にとって明らかであり, 有用であると言える. 従って, 頻度だけでは必ずしもテンプレートとして有用さを測ることはできない. 本稿では, コーパスの全ての文がいずれかのテンプレートによって生成されるという網羅性を仮定し, ドメインの特徴を捉えたテンプレート集合を獲得する手法を提案する.

2 言語の極限同定と極小言語

提案手法は Gold が掲げた言語の極限同定 [6] の枠組みを用いる. テンプレートを空欄を含む文とすると, これは Shinohara が導入した正則パターン [7] と一致し, これを同一視する. 正則パターンを p とするとき, p 中の空欄に語を埋めて生成できる文全体を言語 $L(p)$ と書く. 唯一つの正則パターン p によってコーパス中の全ての文が生成されるとは考えにくいため, 有限個の正則パターン $P = p_1, p_2, \ldots, p_k$ によって生成される正則パターン言語の和集合 $L(P) = L(p_1) \cup \ldots \cup L(p_k)$ の正データがコーパスだと考える (コーパスを文集合 S とするとき $S \subseteq L(P)$). そのような言語和は, 和の数に上限 k を与えた場合に正提示から学習可能であること, そのための効率的な推論アルゴリズムが Arimura らによって示されている [8]. 推論アルゴリズムは正データから高々 k 個の正則パターンを推論結果として出力する. アルゴリズムは高々k 個の正則パターン (集合 P) を仮説としてもっておき, 逐次的に正データを受け取る (集合 S としてプールする) が、仮説と矛盾 ($S \not\subseteq L(P)$) したときに初めて, $MMG(k, S)$ (Algorithm 1) を計算してその出力 P を仮説として更新する. ここで $\top$ は変数一つからなるパターンである. 手続き MMG は次を計算するものである. 自然数 k, 文字列集合 S について $S \subseteq L(P)$ かつ $S \subseteq L(P') \implies P' \not\subset P$ (P の極小性) を満たす P を探索する. ここで P 及び P' は高々k 個の正則パ

ターンからなる集合であって, いずれの真部分集合も S を被覆しないようなものである. 一般に, このような $L(P)$ を極小言語という. 有限の文字列の集合 S

Algorithm 1: $MMG(k, S)$

Input: 自然数 k, 文集合 S
Output: P

```
p ← tighten(⊤, S)
P = {p}
while |P| < k do
    Let p ∈ P
    P' ← division(p, S \ L(P \ p))
    if division succeed then
        P ← (P \ p) ∪ P'
```

が正データとしてあるとき, $MMG(k, S)$ を計算することで, パターンの推論が可能であるが, 我々のタスクに適応するには問題が 3 つある. (1) S のみから適切な k を推定する必要があること. (2) MMG アルゴリズムによって推定されるテンプレート集合には偏りが生じることがあること. この問題は [4] で指摘されている. (3) 対象は英文であり, 出現するトークンをアルファベットとしたが, アルファベットサイズが膨大になり, 計算量的コストが大きい.

(1) について我々は正則パターンの個数 k に上限を与えずに各正則パターンが定める値である抽象度を定義し, これを用いた手法を提案した [3]. 次章では MMG アルゴリズムをベースにして, これらを解決するための改良を述べる.

3 提案手法

正則パターンに, 品詞に対応したクラス変数を導入し拡張する. これはテンプレートに文法的情報をもたせられると同時に, 探索空間の削減に貢献する. 本章ではまず, クラス変数付きの正則パターンの定義を述べ, 次に MMG アルゴリズムをクラス付き正則パターンに適応させた場合を示す. 最後にこれをサブルーチンとして用い推論アルゴリズムを提案する.

3.1 クラス付き正則パターン

アルファベット Σ をシンボルとクラスのペアの有限集合とする. ここでクラスとは $1, 2, \ldots, n$ のいずかれの数である. Σ の要素を (a, i) (a はシンボル, $1 \leq i \leq n$) と書く. 変数集合 X 及びクラス変数集合 $Y_i (1 \leq i \leq n)$ をシンボルの無限集合と定める. ただし, $\Sigma, X, Y_1, \ldots Y_n$ は互いに素とする. $x \in X$ を単に変数と呼び, $y \in Y_i$ をクラス i のクラス変数と呼ぶことにする. $(\Sigma \cup X \cup Y_1 \cup \ldots \cup Y_n)^+$ の要素を **クラス付きパターン** と定義し, クラス付きパターン全体を $\mathcal{CP}$ と書く. クラス付きパターンであって全てのクラス変数及び変数が高々一回出現するものを**クラス付き正則パターン** と定め, クラス付き正則パターン全体を $\mathcal{CRP}$ と書く. 次に, 代入と呼ばれる $\mathcal{CP}$ から $\mathcal{CP}$ への準同型写像を次のように定める. 全て異なる $u_1, u_2, \ldots, u_k \in X \cup Y_1 \cup \ldots \cup Y_n$ をそれぞれ $p_1, p_2, \ldots, p_k \in \mathcal{CP}$ に置換する操作を $[u_1/p_1, u_2/p_2, \ldots, u_k/p_k]$ と書いてこれを代入とする. ただし $u_i \in Y_j$ のとき, p_i はクラス j の単語に限る ($p_i = (*, j) \in \Sigma$). 代入 θ によってパターン p から写されたパターンを $p\theta$ と書く. $p, q \in \mathcal{CP}$ についてある代入 θ があって $q = p\theta$ となるとき $q \preceq p$ と書く. クラス付きパターン p の作る言語を $L(p) = \{s \in \Sigma^* | s \preceq p\}$ で定め, **クラス付きパターン言語** と呼ぶ. クラス付きパターン中に出現するクラス変数及び変数を全て区別することで, クラス付き正則パターンと見做し, $\mathcal{CRP}$ から $\mathcal{CRP}$ へ代入を考えることができる. $p \in \mathcal{CRP}$ に対して $L(p) = \{s \in \Sigma^* | s \preceq p\}$ を **クラス付き正則パターン言語** と呼ぶ.

3.2 MMG アルゴリズムのクラス付き正則パターンへの適応

Arimura らによる MMG をクラス付き正則パターンに適応するには, 基本代入 (basic substitution) を次のようにするだけで良い: $\theta = [x/x_1x_2](x, x_1, x_2 \in X)$, $[x/y](x \in X, y \in Y_i, 1 \leq i \leq n)$, $[y/(a, i)](y \in Y_i, (a, i) \in \Sigma)$. またパターンの偏りを解消するためにサブルーチン *division* [8] を改良する. $division(p, S)$ は次のようなものである. パターン p から基本代入によって得られるパターン全体を $\rho(p)$ としたとき, $division(p, S)$ は $P \subseteq \rho(p)$ かつ $S \subseteq L(P)$ を満たすパターン集合 P であって, 言語 $L(P)$ の包含関係に関して極小な P を計算する. ある k によって $|P| \leq k$ が要請されるため, 実際には言語の集合 $\{L(q) : q \in \rho(p)\}$ から S 全体を被覆するように, できるだけ少ない言語を選択する**集合被覆問題**を解くことで *division* が実現できる. しかし, 選択するパターン数を最小にするだけでは, パターンの偏りの問題が生じる [4]. これを解消するために, パターンに重みを与えた次のような**重み付き集合被覆問題**を解くことを考える. すなわち, S 全体を被覆する最小のパターン集合 $P \subseteq \rho(p)$ の内, $\sum_{p \in P} w(p)$ を最小化する P を計算する. 重み付き集合被覆問題もまた NP-hard であるが実用的な近似ア

ルゴリズムが存在する [5]. 偏りの原因となる極端に具体的なパターン, 極端に汎用的なパターンを避けるように次の 2 つの仮説に基づいてパターンに重み付けをする. パターン $p \in R$, $L(p) \cap S \neq \emptyset$ なる 文字列集合 S について $S' = L(p) \cap S$ とするとき

(1) 具体的過ぎるパターンにはマッチする文が少ないとし, パターンの S に関する部分被覆サイズ $|S'|$ の割合 $s(p, S) = |S'|/|S|$ によって具体的なパターンを避ける.

(2) 文字列 s の bag-of-words 表現 (アルファベットの多重集合) を $bow(s)$ とするとき $c = |\bigcap_{s \in S'} bow(s)| / \max_{s \in S'} |s|$ によって S' のまとまり具合を表現する. 例えば $|S'| = 1$ のとき $c = 1$ である.

S に関するパターン p の重みを s 及び c が共に大きいときに小さい正数であるよう設計し, $w(p, S') = \log s \cdot (\alpha + \log c)$ (α は負の定数) とした. 以上を *division* とし, 変更を加えた MMG アルゴリズムを $MMG(k, S)$ とする. また, $\forall s \in S, s \preceq q$ となるパターン q が与えられたとき, Algorithm 1 でパターン $\top$ から始める (1 行目) 代わりに q から始めても構わない. この手続きを $MMGFrom(S, k, q)$ とする.

3.3 パターンの推論アルゴリズム

以上の MMG アルゴリズムによってクラス付き正則パターン言語の高々k の和集合は学習可能であるが, 先述したようにパターン数 k を決める難しさがある. 本手法では k を決める代わりに各パターンが被覆する文の数について上限 m を与える. すなわち, m より大きな数を被覆するパターンは過汎化だとし, m 以下にのパターンについては変更を加えない. 提案する推論アルゴリズムの擬似コードを Algorithm 2 として示す. m, n, m', n' はパラメータとして設定する自然数である. これは逐次的に正データを受け取り, 受け取った正データにマッチするパターンがあるときだけ, パターンが被覆する文数が m を超えた時だけパターンをさらに MMG によって分割し, マッチするパターンを持たない時は $\overline{S}$ へプールする. プールした文数が一定数 n を超えた時, プールを対象にして MMG を行いパターンを更に加えることをする. MMG に与える文集合 S のサイズは定数 m, n で抑えられるため高速である. 手続き $Add(P, \overline{S}; Q, S')$ は MMG によって生成されたパターン集合 Q を P に追加するが, ここで $|L(q) \cap S'| \geq \ell$ かつ $\forall p \in P, p \not\preceq q$ なる $q \in Q$ のみ P に加える. ここで ℓ は $\ell < m$ なる定数であるが実験では $\ell = 2$ とした. 加えたパターン全体を Q' ($\subseteq Q$) としたとき, 今 P によって被覆されていない文全体は $S' \setminus L(Q')$ であり, これをプール $\overline{S}$ に戻す. 最後に Add は変更を加えた後の $(P, \overline{S})$ を返す.

$RandomSample(T, n)$ は集合 T から要素を n 個, ランダムサンプリングする手続きである. Algorithm 2 で得られた P の各パターンを獲得されたテンプレートだとする.

Algorithm 2: $Infer(S, m, n)$

Input: 文集合 $S = \{s_1, s_2, \ldots, s_T\}$
Output: パターン集合 P

```
1  P ← {}
2  S̄ ← {}
3  S_0 ← {}
4  for t ← 1 to T do
5      S_t ← S_{t−1} ∪ {s_t}
6      if ∃p ∈ P, s_t ⪯ p then
7          S' ← S_t \ L(P \ p)
8          if |S'| > m then
9              Let Q ← MMGFrom(S', m', p)
10             (P, S̄) ← Add(P \ p, S̄; Q, S')
11     else
12         S̄ ← S̄ ∪ {s_t}
13         if |S̄| > n then
14             Let S̄' ← RandomSample(S̄, n)
15             Let Q ← MMG(S̄', n')
16             (P, S̄) ← Add(P, S̄ \ S̄'; Q, S̄')
```

3.3.1 シード

$Infer(S, m, n)$ において, 予めいくつかのパターン $p_1, p_2, \ldots p_k$ を用意し, Algorithm 2 において P の初期値に $\{p_1, p_2, \ldots, p_k\}$ を用いることができ (2 行目), これをシードと呼ぶ. シードはある表現を含むテンプレートの獲得を恣意的に誘導することができる. この場合, アルゴリズムの最後に $L(P \setminus p) \cap S = \emptyset$ なる $p \in P$ を取り除く処理を追加する.

4 実験

あるドメインから獲得したテンプレートはそのドメインの特徴を上手く捉えていることが要請される. そのようなテンプレートはドメインの文には多くマッチし, その他のドメインの文にはほとんどマッチしないはずであり, テキスト分類の素性としても上手く働くことが期待される. そこで我々はテキスト分類実験によって, テンプレートがドメインの特徴を捉えられていることを確認する.

n 個のドメイン $1, 2, \ldots, n$ のそれぞれに属する文集合 $S_1, S_2, \ldots, S_n$ を用意し, S_i から提案手法によってテンプレートの集合 T_i を得たとき, $T_1 \cup T_2 \cup \ldots \cup T_n$ を素性として学習した分類器によって未知文のドメイ

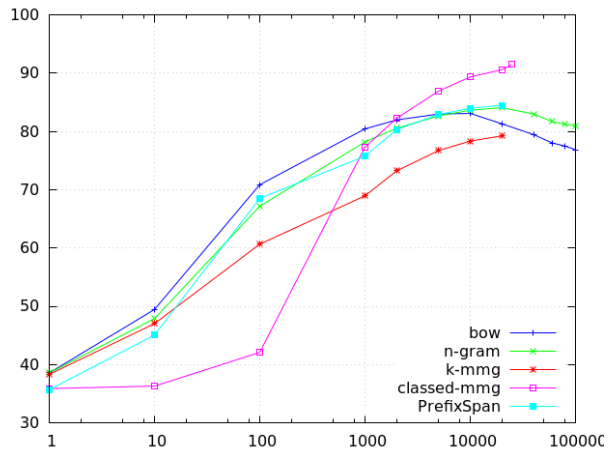

図 1: テキスト分類の結果　横軸:素性数　縦軸:精度

ンを当てるタスクを行う. PubMed [1] から得た論文要旨 (英文) をデータセットとして用いる. この論文要旨には文に BACKGROUND, METHOD, RESULT, CONCLUSION といったラベルがアノテートされている. この 4 つのいずれかのラベルがアノテートされた文を用い, 1 文からその文のラベルを推定するタスクを行う. 他のラベルもあるが数が少ないため, 実験には用いない. 要旨単位で訓練データとテストデータに分割した. 各ラベルの文数は表 1 の通りである. 分類器

表 1: ラベルとその文数

ラベル	訓練データ	テストデータ
BACKGROUND	49,755	2,567
METHOD	59,003	3,014
RESULT	83,369	4,207
CONCLUSION	38,812	1,978

には線形 SVM のフレームワークである LIBLINEAR [2] を用いた. $m = 20, n = 30, m' = m/2, n' = n/2$ とし, *Infer* によって提案手法によるテンプレート集合を各ドメインの文集合から得, それらの和集合を素性として用いた. 比較手法として, 素性だけ異なるものにして学習した分類器を用いる. 素性には, テキスト分類として標準的な bag-of-words 素性, n-gram 素性, 及び, 提案手法同様にパターンを獲得する手法である PrefixSpan を用いて得られたパターンを素性としたものを比較手法とした. n-gram では $n = 1, 2, 3, 4, 5$ を用いた. また素性選択として term frequency でランキングした時の上位 f 件の素性を用いた. PrefixSpan ではパラメータとして頻度の下限に 2000 を用いた. 図 1 は横軸に素性選択をしたときに用いた素性数, 縦軸に分類の精度を用いたプロットである.

[1] http://www.ncbi.nlm.nih.gov/pubmed
[2] https://www.csie.ntu.edu.tw/~cjlin/liblinear/

結果として PrefixSpan 素性は n-gram 素性とほぼ同様の傾向を示した. n-gram 素性は bag-of-words 素性より僅かに良い結果となった. 提案手法は (素性数) > 1000 では常に最良の結果を示した. < 1000 のとき提案手法は他より悪い結果であり, 頻度が大きいテンプレートがいつもドメインの特徴を捉えられるわけではないことを示している.

5　おわりに

本稿ではドメインの特徴を捉えた文テンプレートの獲得手法を提案した. テキスト分類実験では, 獲得した文テンプレートの集合がドメインをよりよく特徴を捉えていることが示された. しかしながら, あくまで集合としての評価であり, 各テンプレートの評価ではなく, 個別のテンプレートを作文支援への有用性という観点から直接評価できたわけではない. また, 個別のテンプレートを自動的に評価し, ランキングすることができれば, 作文支援においてユーザーに示す候補の列挙に役立つと考えられる.

参考文献

[1] E. Filatova, V. Hatzivassiloglou and K. McKeown: Automatic Creation of Domain Templates in *the Proceedings of ACL/COLING, Sydney, Australi, 2006*

[2] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal and M-C. Hsu: PrefixSpan: Mining Sequential Patterns Efficiently by PrefixProjected Pattern Growth in *Proc. 2001 Int. Conf. Data Engineering (ICDE'01)*, pp. 215–224

[3] 服部一浩, 横野光, 相澤彰子: 文書分類のためのフレーズパターンの生成 in *JSAI, 2015*

[4] YK. Ng and T. Shinohara: Inferring Unions of the Pattern Languages by the Most Fitting Covers in *ALT, 2005* pp. 269–282

[5] V. Chvatal: A Greedy Heuristic for Set-Covering Problem in *Mathematics of Operations Research, Vol. 4, No. 3, 1979*, pp. 233–235

[6] E. Gold: Language Identification in the Limit in *Information and Control, 1967*, pp. 447–474

[7] T. Shinohara: Polynomial time inference of pattern languages and its applications in *Proceedings of the 7th IBM Symposium on Mathematical Foundations of Computer Science, 1982*, pp. 191–209

[8] H. Arimura, T. Shinohara, and, S. Otsuki: Finding Minimal Generalizations for Unions of Pattern Languages and its Application to Inductive Inference from Positive Data in *Proc. STACS '94, LNCS, Vol. 775 Springer, Berlin, 1994*, pp. 649–660