アライメント識別モデルを用いた略語定義の自動獲得

岡崎 直観[*1]　辻井 潤一[*1*2*3]

[*1]東京大学大学院情報理工学系研究科　[*2]英国マンチェスター大学
[*3]英国国立テキストマイニングセンター

1. はじめに

略語は，本来の語（完全形）を，短い表現（短縮形）で置き換える言語現象であると同時に，用語の多様性と曖昧性という問題を引き起こしている．例えば，*estrogen receptor* は *ER*，*ESR*，*RE*，*OER* などと略されるが（多様性），テキスト中で単独で現れる *ER* は，*estrogen receptor* 以外にも *endoplasmic reticulum*，*emergency room*，*estradiol receptor* など，指し示す実体が曖昧である．特に，バイオ・医療分野においては，年々数万〜数十万件の略語が生み出されており [Chang 06]，略語は情報検索や固有表現抽出など，言語処理の根幹において重大な問題を引き起こしている [Ananiadou 06, Erhardt 06].

略語が引き起こす用語の多様性・曖昧性の両方の問題を解決するために欠かせない技術が，**略語認識**である．略語認識は，テキスト中に記述されている略語の定義を見つけるために，以下のパターンの括弧表現[*1] に着目する [Schwartz 03].

完全形 '(' 短縮形 ')'　(1)

例えば，次に示す文は括弧の内側に略語の候補 *TTF-1* を含む.

> *We investigate the effect of thyroid transcription factor 1 (TTF-1).*

略語認識の残りのタスクは，括弧の周辺から略語の定義と思われる表現を（もしあれば）抽出することである．このタスクの難しさは，複数の候補から正しい完全形を選ぶところにある．先の例では，*transcription factor 1* と *thyroid transcription factor 1* の両方の表現が，略語 *TTF-1* の文字を同じ順序で含むが，略語の完全形として正しいのは後者である.

略語認識に関する従来研究は，略語に対して正しい完全形を選ぶために，人手で調整されたヒューリスティック [Schwartz 03, Adar 04, Ao 05]，共起頻度 [Hisamitsu 01, Okazaki 06]，機械学習 [Nadeau 05, Chang 06] など，様々なアプローチを提案している．ヒューリスティックに基づく研究は，「完全形と略語の間に存在する余分な語の数」「完全形の語数と略語の文字数の差」などの特徴を手掛かりとし，最適な完全形を選ぶためのルールを設計している．しかし，参照したコーパスにバイアスされたルールになりやすいこと，複数のヒューリスティックを効果的に組み合わせるのが難しいなどの問題を抱えている．統計情報に基づく手法は，ルールに基づく手法と比べて高い精度が期待できるが，低頻度の略語定義を認識するのは，原理上困難である．機械学習に基づく手法は，人手で行っていたルールの最適な統合を自動獲得することを狙っているが，「完全形の語の先頭にある文字が略語を構成する割合」など，略語定義を巨視的に表現する素性を 10〜20 個程度用いるのみで，認識精度において優位性を示すことができなかった.

本研究では，略語認識タスクを，略語とその周辺の表現が与えられたとき，略語の起源（アライメント）を推定する問題として定式化する．略語の周辺に存在する文字が略語の起源である場合，及び起源でない場合の特徴を素性として表現し，最大エントロピー法に基づく識別モデルで多種多様な素性を統合する．識別モデルの学習には，テキスト中で略語と完全形の文字レベルでの対応が付与された「アライメント付きコーパス」を用いる．評価実験では，ヒューリスティックに基づく略語認識，機械学習に基づく略語認識システムと比較し，提案手法による略語認識性能の改善を報告する.

2. 提案手法

2.1 略語アライメント識別モデル

略語候補を含む文を $\boldsymbol{x}$，その文に含まれる略語候補を $\boldsymbol{y}$ とし，これらを文字の並び，$(x_1, ..., x_L)$，$(y_1, ..., y_M)$ で表現する．完全形の文字 x_i が略語の文字 y_j を生じさせる事象を，**文字のマッピング**と呼び，$a = (i, j) \in (\{0, ..., L\} \otimes \{0, ..., M\})$ で表す（ただし $\otimes$ は直積）．マッピング $a = (i, j)$ に関して，$i = 0$ もしくは $j = 0$ であれば，ヌル・マッピングと呼び，$a = (i, 0)$ は x_i が略語 $\boldsymbol{y}$ の構成要素にならなかったこと，$a = (0, j)$ は略語の文字 y_j が完全形 $\boldsymbol{x}$ のどの文字にも由来しないことを示す[*2]．便宜上，$a_{(x)}$ と $a_{(y)}$ で文字マッピングの第 1，第 2 要素を表現する．すなわち，マッピング $a = (i, j)$ に対し，$a_{(x)}$ と $a_{(y)}$ はそれぞれ，i と j に等しい．完全形の文字と略語の文字は，T 個のマッピングから構成される**略語アライメント** $\boldsymbol{a} = (a_1, ..., a_T)$ で対応付けられる．図 1 に，先ほどの例文に含まれる略語 *TTF-1* に対する正解のアライメント（最下行）と，その 2 次元格子表現を示した．略語の文字 't', 't', 'f', '-', '1' は，それぞれ，x_{30}, x_{39}, x_{52},（ヌル・マッピング），x_{59} に由来する.

本研究では，略語を含む文 $\boldsymbol{x}$ と略語候補 $\boldsymbol{y}$ が与えられた時，略語アライメント $\boldsymbol{a}$ の条件付き確率 $P(\boldsymbol{a}|\boldsymbol{x}, \boldsymbol{y})$ を，最大エントロピー法 [Berger 96] に基づき，式 2, 3 で表現する.

$$P(\boldsymbol{a}|\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{Z(\boldsymbol{x}, \boldsymbol{y})} \exp\{\boldsymbol{\Lambda} \cdot \boldsymbol{F}(\boldsymbol{a}, \boldsymbol{x}, \boldsymbol{y})\}, \quad (2)$$

$$Z(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{a} \in C(\boldsymbol{x}, \boldsymbol{y})} \exp\{\boldsymbol{\Lambda} \cdot \boldsymbol{F}(\boldsymbol{a}, \boldsymbol{x}, \boldsymbol{y})\}. \quad (3)$$

ここで，$\boldsymbol{F} = \{f_1, ..., f_K\}$ は K 個のグローバル素性関数で構成されるベクトル，$f_k(\boldsymbol{a}, \boldsymbol{x}, \boldsymbol{y})$ は実数値を返すグローバル素性関数（2.2 節で定義），$\boldsymbol{\Lambda} = \{\lambda_1, ..., \lambda_K\}$ は素性関数に対する重み，$C(\boldsymbol{x}, \boldsymbol{y})$ は与えられた $\boldsymbol{x}$ と $\boldsymbol{y}$ に対して，可能なアラ

*1　節や文を括弧で挿入する場合を取り除くため，括弧の内側の表現が「2 語以内」「2 文字から 10 文字」「少なくても一つの英数字を含む」「先頭の文字が英数字である」という条件をすべて満たす括弧表現のみを，略語認識の対象とする.

*2　本研究では，略語中の記号（' ', '-' などの英数字以外の文字）は，すべてヌル・マッピングにより生成されたと考える.

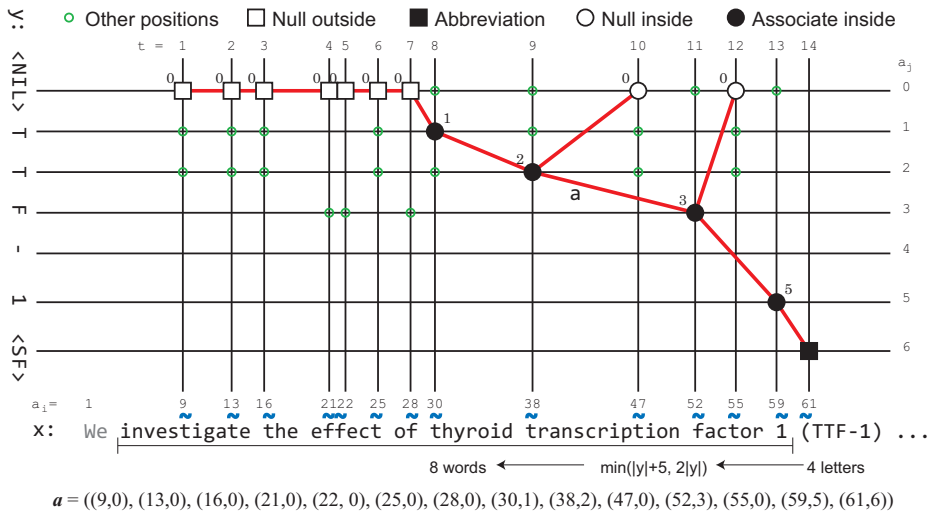

図 1: 例文に対する「正解」の略語アライメント（最下行）と，その 2 次元格子表示.

イメントの集合を返す関数（2.3 節で説明），$Z(\boldsymbol{x}, \boldsymbol{y})$ は条件付き確率分布を正規化する分配関数である.

略語アライメントモデルのパラメータ推定は，通常の最大エントロピー法と同様である．すなわち，N インスタンスの学習セット $\left((\boldsymbol{a}^{(1)}, \boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}), ..., (\boldsymbol{a}^{(N)}, \boldsymbol{x}^{(N)}, \boldsymbol{y}^{(N)})\right)$ が与えられた時，確率モデルの事後確率を最大化する（MAP 推定）．本研究では，過学習を防ぐために L_1 正則化，もしくは L_2 正則化を施し，式 4，もしくは式 5 を最大化する.

$$\mathcal{L}_1 = \sum_{n=1}^{N} \log P(\boldsymbol{a}^{(n)}|\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}) - \frac{||\boldsymbol{\Lambda}||_1}{\sigma_1}, \tag{4}$$

$$\mathcal{L}_2 = \sum_{n=1}^{N} \log P(\boldsymbol{a}^{(n)}|\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}) - \frac{||\boldsymbol{\Lambda}||_2^2}{2\sigma_2^2}. \tag{5}$$

ここで，σ_1 と σ_2 は，それぞれ L_1，L_2 正則化の定数である．式 4 と 5 を最大にする $\boldsymbol{\Lambda}$ は，それぞれ OW-LQN 法 [Andrew 07]，L-BFGS 法 [Nocedal 80] で効率よく求めることが可能である.

略語を認識するタスクは，与えられた $\boldsymbol{x}$ と $\boldsymbol{y}$ に対して，次式を用いて最適なアライメント $\hat{\boldsymbol{a}}$ を求める問題となる.

$$\hat{\boldsymbol{a}} = \underset{\boldsymbol{a} \in C(\boldsymbol{x}, \boldsymbol{y})}{\operatorname{argmax}} P(\boldsymbol{a}|\boldsymbol{x}, \boldsymbol{y}). \tag{6}$$

2.2 素性

完全形が略語の文字を生成する過程では，「大文字である」「単語の先頭に位置する」などの特徴をもつ文字が選ばれやすい．本研究では，位置 t におけるローカル素性 $g_k(\boldsymbol{a}, \boldsymbol{x}, \boldsymbol{y}, t)$ を設計し，グローバル素性関数 $f_k(\boldsymbol{a}, \boldsymbol{x}, \boldsymbol{y})$ の値を次式で求める.

$$f_k(\boldsymbol{a}, \boldsymbol{x}, \boldsymbol{y}) = \sum_{t=1}^{T} g_k(\boldsymbol{a}, \boldsymbol{x}, \boldsymbol{y}, t). \tag{7}$$

本研究で設計したローカル素性は，unigram 素性，bigram 素性に大別される．unigram 素性は，文字マッピング $a_t = (i, j)$ において，完全形の文字 x_i が略語文字 y_j を生成する場合の特徴，もしくは x_i が略語文字として採用されない場合の特徴を表現する．例えば，図 1 において完全形の文字 x_{30} が略語文字 y_1 を生成する事象は，「x_{30} が完全形の語の先頭に位置する」「y_1 が大文字である」「x_{30} と y_1 の両方の文字がそれぞれ語の先頭に位置する」などの特徴が影響していると推測できる．bigram 素性は，隣接する 2 つの文字マッピング a_s と a_t $(1 \leq s < t \leq T)$ に関する特徴を表現する．ここで，位置 t に隣接するマッピングの位置 s は，以下の式で選ぶ.

$$s = \begin{cases} t-1 & \left(a_{t(y)} = 0 \vee \forall u : a_{u(y)} = 0\right) \\ \max\left\{u \;\middle|\; 1 \leq \forall u < t \wedge a_{u(y)} \neq 0\right\} & \text{(それ以外の場合)} \end{cases} \tag{8}$$

式 8 は，位置 t 以前に（ヌルではない）文字マッピングが存在するならば，直近の文字マッピングを選択し，存在しなければ $t-1$ を選択するものである．図 1 において，a_{11}，a_{13} に対する隣接マッピングが a_9，a_{11} であることに注意されたい.

表 1 に，$x_{a_{t(x)}}$, $y_{a_{t(y)}}$, a_t, $x_{a_{s(x)}}$—$x_{a_{t(x)}}$, $y_{a_{s(y)}}$—$y_{a_{t(y)}}$ における特徴を表現する原始素性関数をまとめた．本研究で用いる unigram 素性，bigram 素性は，これらの原始素性関数の組み合わせで表現される．例えば，完全形に含まれる大文字 $x_{a_{t(x)}}$ から，同一の略語の文字 $y_{a_{t(y)}}$ を生成する事象は，次式の unigram 素性で表現できる.

$$g_k(\boldsymbol{a}, \boldsymbol{x}, \boldsymbol{y}, t) = \begin{cases} 1 & \text{x\_ctype}_0(\boldsymbol{a}, \boldsymbol{x}, t) = \texttt{U} \\ & \wedge\ \text{y\_ctype}_0(\boldsymbol{a}, \boldsymbol{y}, t) = \texttt{U} \\ & \wedge\ \text{a\_state}(\boldsymbol{a}, \boldsymbol{y}, t) = \texttt{MATCH} \\ 0 & \text{(それ以外の場合)} \end{cases} \tag{9}$$

表 2 に，原始素性関数を組み合わせる方法を示したテンプレートを示した．従来の機械学習に基づく略語抽出研究と比較すると，提案手法の素性は完全形の文字が略語文字を生成する事象を直接的に表現する点が大きく異なる．また，名詞句が多くを占める略語の完全形の開始／終了位置を認識するために，品詞コードを表す関数 x_pos を導入した．関数 x_diff, x_diff_wd は，連続する 2 つの略語文字が，完全形のどのような位置から生成されるのか，bigram 素性として表現する．さらに，関数 y_diff は，略語の文字の並び（略語の文字と完全形の文字が同じ順序で並んでいる場合は常に 1 を返す）を，素性として表現している.

2.3 候補生成関数

式 3 は与えられた $\boldsymbol{x}$ と $\boldsymbol{y}$ に対して，可能なアライメント集合の和を計算している．$\boldsymbol{x}$ と $\boldsymbol{y}$ の文字数をそれぞれ L と M と

素性抽出関数	返り値
$\mathrm{x\_ctype}_\delta(\boldsymbol{a},\boldsymbol{x},t)$	文字 $x_{a_{t(x)}+\delta}$ が {U (大文字), L (小文字), D (数字), W (空白文字), S (記号) }
$\mathrm{x\_position}_\delta(\boldsymbol{a},\boldsymbol{x},t)$	文字 $x_{a_{t(x)}+\delta}$ の位置が {H (語の先頭), T (後の末尾), S (音素の区切り), I (その他の語の内部), W (空白文字) }
$\mathrm{x\_char}_\delta(\boldsymbol{a},\boldsymbol{x},t)$	文字 $x_{a_{t(x)}+\delta}$ を小文字にしたもの
$\mathrm{x\_word}_\delta(\boldsymbol{a},\boldsymbol{x},t)$	文字 $x_{a_{t(x)}}$ を含む語（オフセット δ 語）を小文字にしたもの
$\mathrm{x\_pos}_\delta(\boldsymbol{a},\boldsymbol{x},t)$	文字 $x_{a_{t(x)}}$ を含む語（オフセット δ 語）の品詞コード
$\mathrm{y\_ctype}_\delta(\boldsymbol{a},\boldsymbol{y},t)$	文字 $y_{a_{t(y)}+\delta}$ が {U (大文字), L (小文字), D (数字), S (記号) }
$\mathrm{y\_position}_\delta(\boldsymbol{a},\boldsymbol{y},t)$	文字 $y_{a_{t(y)}+\delta}$ の位置が略語の {H (先頭), T (末尾), I (内部)}
$\mathrm{y\_char}_\delta(\boldsymbol{a},\boldsymbol{y},t)$	$y_{a_{t(y)}+\delta}$ を小文字にしたもの
$\mathrm{a\_state}(\boldsymbol{a},\boldsymbol{y},t)$	$\{\mathtt{SKIP}\ (a_{t(j)}=0), \mathtt{MATCH}\ (1\le a_{t(j)}\le\lvert\boldsymbol{y}\rvert), \mathtt{ABBR}\ (a_{t(j)}=\lvert\boldsymbol{y}\rvert+1)\}$
$\mathrm{x\_diff}(\boldsymbol{a},\boldsymbol{x},s,t)$	もし2つの文字 $x_{a_{t(x)}}$ と $x_{a_{s(x)}}$ が同じ語に属するなら $(a_{t(x)}-a_{s(x)})$，それ以外なら NONE
$\mathrm{x\_diff\_wd}(\boldsymbol{a},\boldsymbol{x},s,t)$	2つの文字 $x_{a_{t(x)}}$ と $x_{a_{s(x)}}$ の間の存在する語の数
$\mathrm{y\_diff}(\boldsymbol{a},\boldsymbol{y},s,t)$	$(a_{t(y)}-a_{s(y)})$

表 1: $\boldsymbol{x}$ と $\boldsymbol{y}$ において観測される事象を抽出する関数群

素性集合	生成ルール
$\mathrm{x\_state}_\delta(\mathrm{t})$	$\{\mathrm{x\_ctype}_\delta(t), \mathrm{x\_position}_\delta(t), \mathrm{x\_char}_\delta(t), \mathrm{x\_word}_\delta(t), \mathrm{x\_pos}_\delta(t), \mathrm{x\_ctype}_\delta(t)\mathrm{x\_position}_\delta(t), \mathrm{x\_position}_\delta(t)\mathrm{x\_pos}_\delta(t), \mathrm{x\_pos}_\delta(t)\mathrm{x\_ctype}_\delta(t), \mathrm{x\_ctype}_\delta(t)\mathrm{x\_position}_\delta(t)\mathrm{x\_pos}_\delta(t)\}$
x_unigram(t)	$\mathrm{x\_state}_0(\mathrm{t}) \cup \mathrm{x\_state}_{-1}(\mathrm{t}) \cup \mathrm{x\_state}_1(\mathrm{t}) \cup \big(\mathrm{x\_state}_{-1}(\mathrm{t}) \otimes \mathrm{x\_state}_0(\mathrm{t})\big) \cup \big(\mathrm{x\_state}_0(\mathrm{t}) \otimes \mathrm{x\_state}_1(\mathrm{t})\big)$
y_unigram(t)	$\{\mathrm{y\_ctype}_0(t), \mathrm{y\_position}_0(t), \mathrm{y\_ctype}_0(t)\mathrm{y\_position}_0(t)\}$
unigram(t)	$\big[\mathrm{x\_unigram}(t) \cup \mathrm{y\_unigram}(t) \cup \big(\mathrm{x\_unigram}(t) \otimes \mathrm{y\_unigram}(t)\big)\big] \otimes \{\mathrm{a\_state}(t)\}$
trans(s, t)	$\{\mathrm{x\_diff}(s,t), \mathrm{x\_diff\_wd}(s,t), \mathrm{y\_diff}(s,t)\}$
bigram(s, t)	$\big[\big(\mathrm{x\_state}_0(s) \otimes \mathrm{x\_state}_0(t) \otimes \mathrm{trans(s,t)}\big) \cup \big(\mathrm{y\_unigram}(s) \otimes \mathrm{y\_unigram}(t) \otimes \mathrm{trans(s,t)}\big) \cup \big(\mathrm{x\_state}_0(s) \otimes \mathrm{y\_unigram}(s) \otimes \mathrm{x\_state}_0(t) \otimes \mathrm{y\_unigram}(t) \otimes \mathrm{trans(s,t)}\big)\big] \otimes \{\mathrm{a\_state}(s)\mathrm{a\_state}(t)\}$

表 2: unigram 素性，bigram 素性を生成させるルール (引数 $\boldsymbol{a}$, $\boldsymbol{x}$, $\boldsymbol{y}$ は省略).

すると，可能なアライメントの総数は最大で 2^{LM} になるため，式 3 を直接計算するのは非現実的とされてきた．これに対し，動的計画法 [McCallum 05, Blunsom 06]，もしくは n-best アライメントによる近似計算 [Och 02, Liu 05] が提案されている．幸いなことに，略語定義のアライメントでは，略語定義に関する自然な仮定を置くと，考慮すべきアライメント候補集合 $C(\boldsymbol{x},\boldsymbol{y})$ が非常にコンパクトになり，式 3 の直接計算が可能である．以下に，本研究が用いた略語定義に関する仮定を列挙する．紙面の都合により，与えられた $\boldsymbol{x}$ と $\boldsymbol{y}$ に対し，アライメント候補 $C(\boldsymbol{x},\boldsymbol{y})$ を生成するアルゴリズムは省略する．

1. 略語の完全形は，略語以前の $\min(m+5, 2m)$ 語の範囲に存在するものとする [Park 01]．ここで，m は略語中に含まれる英数字の文字数を表す．
2. 略語中のすべての英数字は，完全形の中で（大文字・小文字を無視した）同一の文字に関連付けなければならない．
3. 完全形の複数の文字が一つの略語文字を生成したり，完全形の一つの文字が複数の略語文字を生成してはならない．
4. 完全形の文字から略語を生成するときに，文字の位置を最大で d 回並び変えてよい．ここで「並び替え」とは，並び替え前の（一つのもしくは一連の）文字を選び，別の場所に移動させる操作のことを指す．
5. 仮定 1 で定義されたテキストの領域に，略語の定義が存在するとは限らない．従って，アライメント候補集合は，負のアライメント（すべてヌル・マッピングで構成されたアライメント）を常に含むものとする．

3. 評価

提案手法の有効性を調べるため，既存研究と提案手法の略語抽出性能を比較した．ベースラインとして用いたのは，次に挙げる 5 つの略語抽出システムである: Schwartz & Hearst の方法 (SH) [Schwartz 03], SaRAD [Adar 04], ALICE [Ao 05], Chang & Schütze の方法 (CS) [Chang 06], Nadeau & Turney の方法 (NT) [Nadeau 05]．SH[*3], CS[*4], ALICE[*5] は，ウェブ上で公開されている実装そのもの，SaRAD と NT は我々が論文に基づいて再現したものである．我々が再現した NT システムは，元論文で提案されている素性をすべて実装し，Radial Basis Function（RBF）カーネルの Support Vector Machine（SVM）[*6] をコーパス（後述）から学習し，与えられた略語定義の候補を正例と負例に分類するものである．

本論文で提案したモデルを学習するには，略語文字の起源が明示的に示されたアライメント付き略語コーパスが必要である．我々は，MEDLINE からランダムに選んだ 1,000 件のアブストラクトに対して，手作業で略語定義とアライメントを付与し，1,420 件の括弧表現事例に関して，864（60.8%）件の正例（略語定義）と，556（39.2%）件の負例（その他の括弧表現）を含むコーパスを作成した．提案手法の学習と評価には，10 分割交差検定を用いた[*7]．候補生成関数 $C(\boldsymbol{x},\boldsymbol{y})$ の並べ替え定数 d は，0 または 1 とし，事例あたり平均 9.59（$d=0$）または 154.4（$d=1$）個の完全形候補が生成された．

表 3 に，提案手法とベースラインシステムの略語認識の適合率（P），再現率（R），F1 スコア（F1）をまとめた．提案手法（$d=0$; L_1 正則化）が最も良い F1 スコア（0.971）を収めた．略語文字の並び替えを考慮した場合（$d=1$）は，再現率が最大（0.975）となったが，考慮する完全形の候補が増えたために，適合率が低下した．

表 4 は，異なる素性セットを用いた時の提案手法の F1 スコアを示したものである．基本素性セット（1）は，x_position と x_ctype の組み合わせのみで素性を構成するが，0.905 の F1 ス

*3 Abbreviation Definition Recognition Software: http://biotext.berkeley.edu/software.html
*4 Biomedical Abbreviation Server: http://abbreviation.stanford.edu/
*5 Abbreviation LIfter using Corpus-based Extraction: http://uvdb3.hgc.jp/ALICE/ALICE_index.html
*6 LIBSVM – A Library for Support Vector Machines: http://www.csie.ntu.edu.tw/~cjlin/libsvm/
*7 パラメータ推定時の正則化定数は，$\{0.1, 0.2, 0.3, 0.5, 1, 2, 3, 5, 10\}$ を実験的に試し，$\sigma_1=\sigma_2=3$ と決定した．

システム	P	R	F1
Schwartz & Hearst (SH)	**.978**	.940	.959
SaRAD	.891	.919	.905
ALICE	.961	.920	.940
Chang & Schütze (CS)	.942	.900	.921
Nadeau & Turney (NT)	.954	.871	.910
Proposed ($d = 0$; L_1)	.973	.969	**.971**
Proposed ($d = 0$; L_2)	.964	.968	.966
Proposed ($d = 1$; L_1)	.961	.974	.968
Proposed ($d = 1$; L_2)	.957	**.975**	.966

表 3: 略語抽出システムの適合率・再現率・F1 スコア

#	原始素性関数群	F1
(1)	x_position + x_ctype	.905
(2)	(1) + x_char + y_char	.885
(3)	(1) + x_word + x_pos	.941
(4)	(1) + x_diff + x_diff_wd + y_diff	.959
(5)	(1) + y_position + y_ctype	.964
(6)	すべての原始素性関数を用いた場合	.966

表 4: 異なる素性を用いた時の F1 スコアの比較（$d = 0$; L_2）

コアを達成している．分類器に素性を追加することにより，概ね F1 スコアは向上していくが，x_char と y_char（2）は逆に F1 スコアを下げる結果となった．これは，略語アライメントモデルが，学習コーパスに含まれる特定の文字の存在に頼り，過学習を起こすためであると考えられる．興味深いことに，提案手法は 4 つの原始素性関数を組み合わせることで，高い F1 スコアを達成できることが分かる（設定 5）．

表 5 に，提案手法（$d = 0$）が生成した 2,489,690 個の素性のうち，L_1 正則化による MAP 推定が高い重みを割り当てた 10 個の素性を挙げた．unigram 素性と bigram 素性は，それぞれ “U:” と “B:” のプレフィックスで始まり，位置 s における条件（bigram 素性のみ），位置 t における条件，マッピング（M）もしくはヌル・マッピング（S）が，記号 ‘/’ で区切られて続く．例えば，1 番目の素性は，完全形の語の先頭の文字を大文字に変換し，略語の先頭の文字を生成させる事象を表現している．4 番目の素性は，完全形の同じ語に含まれる 2 つの小文字から，連続する略語文字を生成することを勧めている．人手でこれらのルールを発見・統合することは難しく，略語の生成過程を考察する上で，興味深い結果と言えよう．

4. 結論

本論文では，略語認識の新しいアプローチとして，略語アライメント識別モデルを提案した．評価実験では，提案手法が既存の研究よりも認識精度で上回ることを示した．略語定義のアライメントが付与された文を学習コーパスとして用い，完全形の文字が略語を生成する事象を細かく記述する素性を獲得した．今後の課題は，並び替えが施された略語の認識向上，文字の一致が成立しない略語（例えば *‘deficient’* が *‘-’* に略される）への対応，一般的な言語パターンへの拡張（例えば *aka*, *abbreviated as*）などが考えられる．さらに，アライメントが付与されていない略語コーパスから，アライメントを自動的に推定しながら，学習を進める方法を探究したいと考えている．

謝辞

本研究を進めるにあたり，科学技術振興調整費・重要課題解決型研究等の推進「日中・中日言語処理技術の開発研究」の支援を受けた．

素性	重み（λ）
U: x_position$_0$=H;y_ctype$_0$=U;y_position$_0$=H/M	1.73699
B: y_position$_0$=I/y_position$_0$=I/x_diff=1/M-M	1.34695
U: x_ctype$_{-1}$=L;x_ctype$_0$=L/S	0.963417
B: x_ctype$_0$=L/x_ctype$_0$=L/x_diff_wd=0/M-M	0.940087
U: x_position$_0$=I;x_char$_1$=‘t’/S	0.916447
U: x_position$_0$=H;x_pos$_0$=NN;y_ctype$_0$=U/M	0.867857
U: x_ctype$_{-1}$=S;x$_c$type$_0$=L;M	0.864736
B: x_char$_0$=‘o’/x_ctype$_0$=L/y_diff=0/M-S	0.712617
U: x_char$_{-1}$=‘o’;x_ctype$_0$=L/M	0.697641
B: x_position$_0$=H/x_ctype$_0$=U/y_diff=1/M-M	0.664182

表 5: 高い重みが割り当てられた 10 個の素性（$d = 0$; L_1）

参考文献

[Adar 04] Adar, E.: SaRAD: A Simple and Robust Abbreviation Dictionary, *Bioinformatics*, Vol. 20, No. 4, pp. 527–533 (2004)

[Ananiadou 06] Ananiadou, S., Kell, D. B., and Tsujii, ichi J.: Text mining and its potential applications in systems biology, *Trends in Biotechnology*, Vol. 24, No. 12, pp. 571–579 (2006)

[Andrew 07] Andrew, G. and Gao, J.: Scalable training of L1-regularized log-linear models, in *Proceedings of ICML 2007*, pp. 33–40 (2007)

[Ao 05] Ao, H. and Takagi, T.: ALICE: An Algorithm to Extract Abbreviations from MEDLINE, *Journal of the American Medical Informatics Association*, Vol. 12, No. 5, pp. 576–586 (2005)

[Berger 96] Berger, A. L., Pietra, V. J. D., and Pietra, S. A. D.: A maximum entropy approach to natural language processing, *Computational Linguistics*, Vol. 22, No. 1, pp. 39–71 (1996)

[Blunsom 06] Blunsom, P. and Cohen, T.: Discriminative word alignment with conditional random fields, in *Proceedings of ACL 2006*, pp. 65–72 (2006)

[Chang 06] Chang, J. T. and Schütze, H.: Abbreviations in Biomedical Text, in Ananiadou, S. and McNaught, J. eds., *Text Mining for Biology and Biomedicine*, pp. 99–119, Artech House, Inc. (2006)

[Erhardt 06] Erhardt, R. A.-A., Schneider, R., and Blaschke, C.: Status of text-mining techniques applied to biomedical text, *Drug Discovery Today*, Vol. 11, No. 7–8, pp. 315–325 (2006)

[Hisamitsu 01] Hisamitsu, T. and Niwa, Y.: Extracting useful terms from parenthetical expression by combining simple rules and statistical measures: A comparative evaluation of bigram statistics, in Bourigault, D., Jacquemin, C., and L’Homme, M.-C. eds., *Recent Advances in Computational Terminology*, pp. 209–224, John Benjamins (2001)

[Liu 05] Liu, Y., Liu, Q., and Lin, S.: Log-linear models for word alignment, in *Proceedings of ACL 2005*, pp. 459–466 (2005)

[McCallum 05] McCallum, A., Bellare, K., and Pereira, F.: A Conditional Random Field for Discriminatively-trained Finite-state String Edit Distance, in *Proceedings of UAI 2005*, pp. 388–395 (2005)

[Nadeau 05] Nadeau, D. and Turney, P. D.: A Supervised Learning Approach to Acronym Identification, in *8th Canadian Conference on Artificial Intelligence (AI’2005) (LNAI 3501)*, p. 10 pages (2005)

[Nocedal 80] Nocedal, J.: Updating Quasi-Newton Matrices with Limited Storage, *Mathematics of Computation*, Vol. 35, No. 151, pp. 773–782 (1980)

[Och 02] Och, F. J. and Ney, H.: Discriminative training and maximum entropy models for statistical machine translation, in *Proceedings of ACL 2002*, pp. 295–302 (2002)

[Okazaki 06] Okazaki, N. and Ananiadou, S.: Building an abbreviation dictionary using a term recognition approach, *Bioinformatics*, Vol. 22, No. 24, pp. 3089–3095 (2006)

[Park 01] Park, Y. and Byrd, R. J.: Hybrid text mining for finding abbreviations and their definitions, in *Proceedings of EMNLP 2001*, pp. 126–133 (2001)

[Schwartz 03] Schwartz, A. S. and Hearst, M. A.: A simple algorithm for identifying abbreviation definitions in biomedical text, in *Pacific Symposium on Biocomputing (PSB 2003)*, No. 8, pp. 451–462 (2003)