Conditional Random Fields を用いた略語抽出

岡崎　直観 †‡　　辻井　潤一 †§¶

† 東京大学大学院情報理工学系研究科
‡ 独立行政法人 科学技術振興機構 (JST)
§ School of Informatics, University of Manchester, UK
¶ National Centre for Text Mining, UK

1　はじめに

生物・医学分野では，遺伝子，たんぱく質，化学化合物，薬品，有機体などで使用される用語の数が，飛躍的に増加している．既存の用語資源や科学データベース（例えば Swiss-Prot[1]，UMLS[2]，SGD[3]）は，この増加のスピードに対応できず [11]，生物・医学分野の文献を扱う上で，用語管理が重要な役割を担っている [2, 5]．特に，略語は活発に生成されており，2004 年には約 64,242 件の略語が新たに生成されたと推定されている [3]．

略語は，遺伝子，たんぱく質などの重要なエンティティを表現すると同時に，用語の多様性と曖昧性の問題を増大させる．例えば，*estrogen receptor* に言及している文を収集し，情報抽出を行う場合を考える．*estrogen receptor* は，*ER*, *ESR*, *OER*, *RE* 等と略されることが多いので，これらの略語を含む文にも着目し，再現率を向上させたい．一方で，*ER* は *endoplasmic reticulum*, *emergency room* などの全く異なるエンティティの略でもあるので，「*estrogen receptor* の意味で用いられている *ER*」だけを選別し，精度の低下を防ぐ必要がある．

略語の多様性と曖昧性の問題を解決する第一歩として，略語とその定義の関係を認識する必要がある．これまで，生物・医学文献において略語とその定義を自動的に抽出する方法として，ルールに基づく手法 [1, 12]，統計情報に基づく手法 [6, 9]，機械学習に基づく手法 [3, 8] が提案されている．シンプルなルールに基づく手法 [12] は，略語のすべての文字が含まれる最短の表現を，略語の定義として抽出する．ルールに基づく手法は，略語定義の候補に含まれる文字が略語に偶然一致してしまう場合や，略語における文字の並びが定義と異なる場合に，うまく対応できないことが指摘されている [9]．統計情報に基づく手法は，略語とその定義の共起度を用いて略語抽出を行う．ルールに基づく手法と比べて高い精度が期待できるので，略語の辞書を作る目的に適しているが，文書内のみで通用する低頻度の略語定義を認識するのは，原理上困難である．

機械学習に基づく従来手法は，ルールに基づく手法で用いられるスコア関数やルールを，学習で統合するものである．Chang ら [3] は，与えられた略語定義の候補に対し，定義らしさを評価する 8 つの素性値を計算し，ロジスティック回帰モデルで統合した．素性値には，「定義において語の先頭にある文字が略語を構成する割合」「定義において音素の先頭にある文字が略語を構成する割合」「定義に含まれる語が略語の構成に関与する割合」などが用いられる．Nadeau ら [8] は，同様に 17 種類の素性値を，Support Vector Machine (SVM) や決定木などで統合した．しかし，これまでの機械学習に基づく従来手法は，用いられている素性の数が少なく，素性の設計も恣意的であった．このため，機械学習を用いても，ルールに基づく手法と同程度の精度しか達成できず，略語認識における従来からの問題点を解決するに至らなかった．

本研究では，生物・医学文献に含まれる略語の定義を認識する「タガー」を構築する．すなわち，略語の周辺の表現に着目し，略語の定義と思われる文字列の並びと，略語の定義としてはふさわしくない文字の並びを識別するモデルを獲得する．識別モデルとして，条件付き確率場（CRFs: Conditional Random Fields）を採用し，柔軟な素性設計を行う．略語認識に用いる訓練事例は，手作業で作成した略語定義リストと，MEDLINE アブストラクトを用い，半自動的に大量に獲得する．

[1] http://www.ebi.ac.uk/swissprot/
[2] http://umlsinfo.nlm.nih.gov/
[3] http://www.yeastgenome.org/

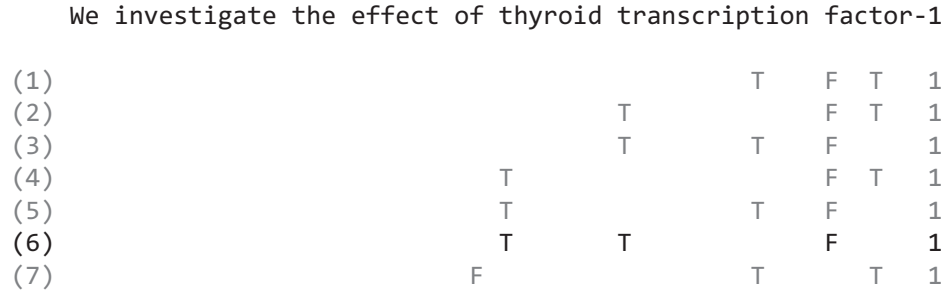

図 1: 略語周辺表現への文字割り当てによる略語定義認識

2 提案手法

本研究における略語認識タスクは，入力テキスト（文など）が与えられた時，その中に含まれる略語（短縮形）と定義（完全形）を,（もし存在するならば）すべて見つけることと定義する．このタスクを,「略語認識」と「定義認識」の 2 つのステップに分解する．

略語認識では，入力テキスト中で略語とその定義が併記されていると思われる個所を見出す．従来研究は，以下に示す括弧表現を手がかりとして，略語（*short form*）を抽出している．

long form ‘(’ *short form* ‘)’ (1)

本研究では，Schwartz ら [12] の手法と同様に，括弧内のフレーズが,「2 語以下」,「2〜10 文字で構成」,「1 文字以上のアルファベットを含む」,「先頭の文字がアルファベットまたは数字」という条件を満たす場合のみ,そのフレーズを略語の候補とする．

定義認識ステップでは，略語とその周辺の表現が与えられた時，その周辺の表現から略語の構成要素を見つけることができるか検討する．図 1 は，略語周辺の表現に略語の文字を割り当て，略語 *TTF-1* の定義を認識する例を示す．例えば，略語 *TTF-1* の定義の候補として，(3) は *transcription factor 1* を表している．この候補は誤りであり，正しい定義は (6) の *thyroid transcription factor 1* である．また，図 1 には挙げていないが，*the* や *effect* の文字 ‘t’ が略語を構成する可能性もある．このように，略語定義認識は，複数存在する文字の割り当て方の中から，正しい文字割り当てを選ぶか（略語の定義が抽出される），すべての文字割り当てを棄却する（略語の定義は抽出されない）ことで遂行できる．以降では，略語定義への文字割り当てタスクを，系列ラベリング問題として定式化する．

2.1 系列ラベリング問題による略語定義認識

これまでの議論から，本研究の目標は，略語周辺の文字列 $\boldsymbol{s} = (s_1, ..., s_L)$ に対し，最適な略語の文字割り当てラベル系列 $\boldsymbol{d} = (d_1, ..., d_L)$ を求めることである．もじ，文字 s_i が略語を生成させた文字でない場合は，ラベル $d_i =$ ‘*’ を割り当てる．この問題をそのまま $\boldsymbol{s}$ から $\boldsymbol{d}$ へのラベリング問題として定式化すると，ラベル系列 $\boldsymbol{d}$ には大量の ‘*’ ラベルが含まれることになり，通常の CRF （線形 CRF など）では，ラベル同士の依存関係をうまく反映できない．また，文字で表現されたラベルの依存関係（例えば ‘H’ を割り当てた直後に ‘H’ を割り当てやすい）を学習しても，ラベル空間がスパースになり，学習の効果が期待できない．

そこで，略語周辺の文字列 $\boldsymbol{s}$ から，略語文字割り当てに関与すると思われる部分だけを抜き出し，入力記号系列 $\boldsymbol{x} = (x_1, ..., x_T)$ を生成する．先行研究 [10, 12] を参考に，略語 a（文字数 $|a|$）の定義は，括弧表現の $\min(|a| + 5, 2|a|)$ 単語前から，開き括弧の直前までの範囲に存在すると仮定する．この範囲の単語列の中で，略語を構成する文字と単語区切りが存在する個所を抜き出して，入力記号系列 $\boldsymbol{x}$ を作成する．図 2 の例では，文字数 4 の略語[4]*TTF-1* の括弧表現よりも前の 8 単語[5]に着目し，文字 ‘t’, ‘f’, ‘1’ と単語区切りを抽出して入力記号系列 $\boldsymbol{x}$ を生成している．ただし，記号 ‘_’ はスペースやハイフンなどの単語区切りを示す．

このようにして得た入力記号系列 $\boldsymbol{x}$ に対して，出力ラベル系列 $\boldsymbol{y} = (y_1, ..., y_T)$ を割り当てる．用いる出力ラベル集合を，表 1 に示す．略語の定義が始まる箇所には，通常 `B1m` ラベルが割り当てられる．これは,「略語の先頭（1 番目）の文字にマッチして，略語が定義されている領域が開始された」ことを示す．もし，*beta2*

[4]略語の文字数は，英数字の文字数と定義し，ハイフンやスペースなどは文字数に含めない．
[5]ハイフンは単語の区切りとしてみなす．

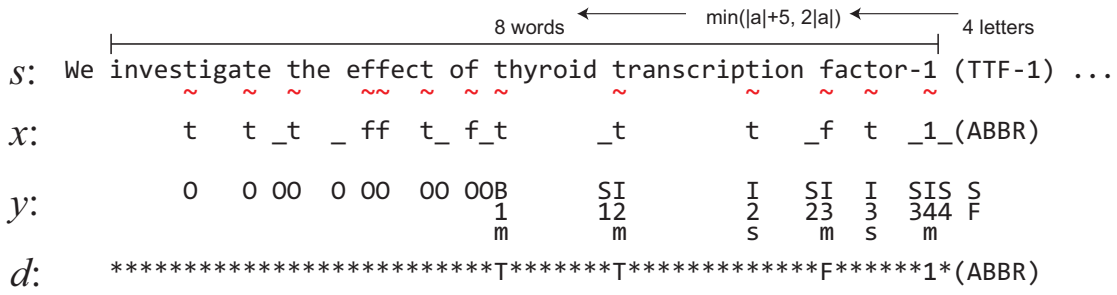

図 2: 略語定義ラベリング

表 1: ラベルの集合

ラベル	内容	例
Bim	略語定義が略語 i 番目の文字にマッチして開始された	B1m
Iim	略語 i 番目の文字にマッチした（略語定義は継続される）	I2m
Iis	略語 i 番目の文字にマッチした後に現れた文字をスキップした（略語定義は継続される）	I2s
Si	略語 i 番目の文字がマッチした後にスペースが現れた（略語定義は継続される）	S2
SF	略語自体	SF
O	略語定義の外側	O

adrenergic receptor (ADRB2) のように，略語と定義の語順が異なる場合，B4m ラベルが略語定義の開始を示すことになる．

略語定義の中で文字が略語にマッチした時には，I2m, I3m などのラベルを割り当てる．例えば，I2m は「略語の 2 番目の文字にマッチして，略語定義の領域が継続された」ことを意味する．このように，略語を構成する元となった表現には，ポストフィックス m を持つラベルが割り当てられる．図 2 の例では，*thyroid transcription factor 1* の下線で記した個所に，それぞれ B1m, I2m, I3m, I4m のラベルが割り当てられる．ポストフィックス m を持つラベルが割り当てられた個所から，元々の記号を抜き出し，他の部分を割り当てなし（‘*’）で埋めると，略語の文字割り当てラベル系列 $\boldsymbol{d}$ が得られる．

transcription の下線で示した箇所のように，略語の文字 ‘t’ が存在するにもかかわらず，略語を構成しているとは見なせないケースがある．このときは，ポストフィックス s を持つラベル（I2s など）を割り当て，マッチした場合と区別する．同様に，略語定義中で単語区切り（スペースやハイフンなど）があったときは，S1, S2 等のラベルを割り当てる．略語定義の外側にある文字には，すべて O ラベルを割り当てる．略語自身が出現する個所には，SF というラベルを割り当て，出力ラベル系列を終端する．

このラベル集合の定義により，略語抽出の従来研究で用いられていたヒューリスティックやルールが，以下のように自然に取り込まれる．

略語にマッチする文字　従来研究では,「語の先頭の文字は略語を構成しやすい」「大文字は略語を構成しやすい」などのヒューリスティックを用いて，略語定義抽出ルールやスコア関数を設計していた．本研究では，入力記号列 $\boldsymbol{x}$ とマッチを表すラベル（B1m, I2m）を結びつける素性関数を定義することにより，これらのヒューリスティックを柔軟にモデル化することができる．また，入力記号列の $\boldsymbol{x}$ のどのような特徴を持つ文字が，範囲外（O）やスキップ（I2s）ラベルを取るのか，同様に素性関数で表現できる．

略語定義の語数　従来研究では,「略語定義の語数は略語の文字数以下が望ましい」「略語定義の中に含まれる略語に関与しない語の数は少ない方がよい」などがルールやスコア関数として用いられる．本研究では，出力ラベル系列にスペースラベル（S1, S2 など）を取り入れ，略語定義中に含まれる語の数が遷移素性や状態素性で表現される．

略語定義の開始・終了位置　従来研究では「略語定義は括弧表現の直前で終わるのが望ましい」というルールが用いられる．本研究において，文字数 4 の略語の

定義を認識する場合，出力ラベルが I4m → S4 → SF と遷移して終端されれば，上述のルールを満たしていることになる．もし，略語定義と略語の間に余分な表現（*gene, method* 等）が割り込む場合は，I4m と SF の間に，余分なスペースラベル S4，もしくはスキップラベル I4s が挿入されることになる．このように，略語定義の開始位置と終了位置は，出力ラベルに関連付けられた遷移素性や状態素性で表現される．

略語文字の並び順　略語定義の語順と略語の文字の順序が一致するときには，出力ラベル系列は B1m, I2m, I3m のように，1 から順に増加するラベルが割り当てられる．*beta2 adrenergic receptor (ADRB2)* のように，略語と定義の語順が異なる場合でも，B4m, I5m, I1m, I2m, I3m のように，$4 \rightarrow 5, 1 \rightarrow 3$ という 2 分割された増加関係が現れる．このような略語の文字の並び順に関する情報は，遷移素性として学習器に取り込まれる．

2.2 Conditional Random Fields

本研究は，入力記号列 $\boldsymbol{x}$ と出力ラベル系列 $\boldsymbol{y}$ の対応関係を条件付き確率場 (CRFs: Conditional Random Fields) [7, 13] でモデル化する．CRF は入力記号列 $\boldsymbol{x}$ と出力ラベル系列 $\boldsymbol{y}$ の条件付き確率分布 $P(\boldsymbol{y}|\boldsymbol{x})$ を式 2 で定義する．

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

ここで，関数 $\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x})$ は大域素性ベクトル，$\boldsymbol{\Lambda} = (\lambda_1, ..., \lambda_K)$ は素性集合の重みを表すベクトルで，λ_k は素性 $\boldsymbol{F}_k(\boldsymbol{y}, \boldsymbol{x})$ の重みに対応する．$Z(\boldsymbol{x})$ は分配関数で，式 3 で与えられる．

$$Z(\boldsymbol{x}) = \sum_{\boldsymbol{y}} \exp\{\boldsymbol{\Lambda} \cdot \boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x})\} \tag{3}$$

CRF の学習は，N 件のインスタンスから構成される学習データ $\left((\boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}), ..., (\boldsymbol{x}^{(N)}, \boldsymbol{y}^{(N)})\right)$ が与えられたとき，条件付き確率分布の対数尤度，

$$\mathcal{L}_\Lambda = \sum_{i=1}^{N} \log P(\boldsymbol{y}^{(i)}|\boldsymbol{x}^{(i)}), \tag{4}$$

を最大にする素性重みベクトル $\boldsymbol{\Lambda}$ を求める問題に帰着する．素性重みベクトルは Gaussian 事前確率分布に従うと仮定すると，式 4 が最大になる条件は次式で与えられる．

$$\frac{\partial \mathcal{L}_\Lambda}{\partial \boldsymbol{\Lambda}} = \sum_{i=1}^{N} \boldsymbol{F}(\boldsymbol{y}^{(i)}, \boldsymbol{x}^{(i)}) - \sum_{i=1}^{N} \sum_{\boldsymbol{y}^{(i)}} P(\boldsymbol{y}^{(i)}|\boldsymbol{x}^{(i)}) \boldsymbol{F}(\boldsymbol{y}^{(i)}, \boldsymbol{x}^{(i)}) - \frac{\boldsymbol{\Lambda}}{\sigma^2} = 0 \tag{5}$$

式 5 の第 1 項は素性の学習データにおける出現頻度，第 2 項は素性の確率モデルにおける出現頻度 (期待値)，第 3 項は事前確率分布によるペナルティ項である．CRF の学習の大半は，第 2 項を計算することに費やされるが，線形マルコフ連鎖モデルを採用する場合は，forward-backward アルゴリズムを用いて効率よく計算できる．式 5 の制約を満たす素性重みベクトル $\boldsymbol{\Lambda}$ は，準ニュートン法である L-BFGS 法 [4] で求める．学習で獲得した確率分布関数 $P(\boldsymbol{y}|\boldsymbol{x})$ を用いると，与えられた入力記号列 $\boldsymbol{x}$ に対する最適な出力ラベル系列 $\hat{\boldsymbol{y}}$ は，次式で与えられる．

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{x}) \tag{6}$$

線形マルコフ連鎖モデルの場合，式 6 は Viterbi アルゴリズムで効率よく求められる．

2.3 素性

表 2 は，与えられた学習インスタンス中の入力記号 x_t と出力ラベル y_t から，素性関数を生成させるためのテンプレートである．例えば，図 2 が学習インスタンスとして与えられ，*thyroid* の文字 't' の個所から「大文字」に関する素性を生成すると，次のような関数になる．

$$f(x_t, y_t) = \begin{cases} 1 & (x_t \text{ が大文字 \& } y_t \text{ が 'B1m'}) \\ 0 & (\text{それ以外の場合}) \end{cases} \tag{7}$$

テンプレート中のオフセットは，位置 x_t の前後の表現を素性に反映するためのもので，オフセットが c_{-1}, c_0, c_{+1} のときは，位置 x_t の前後の文字からも素性を作成する．同様に，オフセットが w_{-1}, w_0, w_{+1} の場合は，前後の単語からも素性を作成する．

今回設計した素性は，従来研究で用いられているヒューリスティックをほぼ網羅している．例えば,「大文字・小文字」素性は,「大文字で記述される文字は略語を構成要素となりやすい」というヒューリスティックを反映している．また,「文字の相対位置」素性は，単語

表 2: 素性生成テンプレート

素性	オフセット	内容	例
文字	c_{-1}, c_0, c_{+1}	文字 c を小文字に変換したもの	ch:-1:' ', ch:0:'t', ch:+1:'r'
大文字・小文字	c_{-1}, c_0, c_{+1}	文字 c が大文字であるか（upper），小文字であるか（lower）	lower:-1, upper:0, lower:+1
文字種別	c_{-1}, c_0, c_{+1}	文字 c が英文字であるか（alpha），数字であるか（digit），記号であるか（symbol）	symbol:-1, alpha:0, alpha:+1
文字の相対位置（語）	c_{-1}, c_0, c_{+1}	文字 c が語 w の先頭であるか（w-head），末尾であるか（w-tail），語の内部であるか（w-inner）	w-head:0, w-inner:+1
略語文字との一致	c_{-1}, c_0, c_{+1}	文字 c が略語の p 番目の文字と一致する（match=p）	match=1:0, match=2:0
単語	w_0	単語 w を小文字に変換したもの	wd:0:thyroid
品詞	w_{-1}, w_0, w_{+1}	単語 w の品詞タグ	pos:-1:IN, pos:0:NN, pos:+1:NN
遷移素性	—	ラベルの bigram	O--B1m, B1m--S1

の先頭や音素の先頭にある文字と略語を関連付けるものである．

「略語文字との一致」素性は，入力記号 x_t と略語の文字を直接対応付けるものである．例えば「x_t が略語の 2 番目の文字と同じであれば，y_t は I2m になりやすい」という関係を表現する．また，オフセット $+1$ の素性は「x_t の次の文字が略語の 2 番目の文字と同じであれば，y_t は B1m になりやすい」という関係を表現できる．これは，*adriamycin (ADM)* のように，定義の連続した文字がそのまま略語を構成する場合に有効である．「単語」「品詞」素性は，定義中の単語や品詞の情報から略語定義の開始位置，終了位置を推定するものである．これにより,「略語の定義が前置詞から始まることは稀」などのルールが自動的に獲得できる．

2.4 ラベル系列に関する制約

ところで，式 6 で推定された出力ラベル系列 $\hat{\boldsymbol{y}}$ が，略語定義として有効な候補を表現するとは限らない．例えば，出力系列 $\hat{\boldsymbol{y}}$ 中にラベル Bm1 が複数回出現したり，I2m の直後に S3 ラベルが現れる場合などは，条件付き確率が最大になるとしても，略語定義の候補としては不適合である．これは，略語定義認識タスクでは，出力ラベル系列内に強い依存関係が存在し，ラベルの独立性やマルコフ性が薄いためである．言い換えれば，ある位置 t におけるラベル付けは，過去 $(1, ..., t-1)$ におけるラベル付け結果に強く依存するためである．

本研究では，この問題に対処するため，与えられた入力記号系列 $\boldsymbol{x}$ に対し，略語定義候補として適合する出力ラベル系列（ラティス）をすべて列挙する．すなわち，学習を行う時は，学習インスタンスの記号系列 $\boldsymbol{x}^{(i)}$ それぞれに対し，可能なラベル系列集合 $Y^{(i)}$ を求めておく．ラベル系列のマルコフ性を仮定しないので，式 5 において forward-backward アルゴリズムを使うことはできない．代わりに，列挙しておいた出力ラベル系列集合 $Y^{(i)}$ を用い，式 5 を定義の通り計算する．

一般的に，可能な出力ラベル系列の数 $|Y^{(i)}|$ は，ラベルの種類数を L，学習インスタンス $(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})$ における入出力系列の長さを T とすると，L^T である．すなわち，式 5 の $\sum_{\boldsymbol{y}^{(i)} \in Y^{(i)}}$ の計算回数は，入出力系列の長さ T に対して指数的に増大するため，通常はすべての出力系列の条件付き確率 $P(\boldsymbol{y}^{(i)}|\boldsymbol{x}^{(i)})$ を直接計算することはない．しかし，入力記号系列 $\boldsymbol{x}^{(i)}$ に対して，略語定義として適合するラベル系列集合の数 $|Y^{(i)}|$ が小さい場合は，式 5 を直接計算することが可能である．

3 実験

3.1 学習データの獲得

略語認識で必要な学習データは，図 2 に示されていた $\boldsymbol{s}$ と $\boldsymbol{d}$ のように，略語周辺の文字列に対して文字割り当てラベル系列が付与されたテキストである．しかし，略語が定義されているテキストに対して，図 2 で示したアノテーション作業を人手で行うのは，大変な時間と労力が必要である．本研究では，略語の定義と文字割り当て情報（アライメント）が付与された特殊な略語定義リストを作成し，その略語を含む MEDLINE アブストラクトを収集することで，大量の学習データを半自動的に獲得した．

図 3 に，今回作成した略語定義リストの例を示した．このリストには，略語 *ER* の MEDLINE アブストラクトにおけるすべての定義と，それぞれの定義における略語の構成要素（アライメント）が示されている．例えば，図 3 の 2–3 行目は，*early ratio* という略語の定義において，語 *early* の先頭の 'e' が略語の 1 番目の文字に対応し，語 *rate* の先頭の 'r' が略語の 2 番目に対応していることを表している．

このように，特定の略語に対し，その定義とアライメントを MEDLINE アブストラクトからすべて列挙す

```
# ER
early ratio
1     2
estrogen receptor
1         2
ethoxyresorufin
1     2
etoposide-resistant
1          2
7-ethoxy-resorufin
  1      2
receptors for estrogen
2              1
...
```

図 3: 略語 *ER* の定義アライメント

```
contents of estradiol receptor (ER)
    *       E  *      R* *   *
levels of estrogen receptor (ER)
 * *      E  *  *  R* *   *
reference to estrogen receptor (ER)
** ***  *    E  *  *  R* *   *
Nuclear estrogen receptor (ER)
    * * E  *  *  R* *   *
Ten were estrogen receptor (ER)
 *   *** E  *  *  R* *   *
in the oestrogen receptor (ER)
     *  E  *  *  R* *   *
(PI) and nuclear receptors (ER)
            * * ** *   *
A radial stiffness modulus (ER)
  *            *
...
```

図 4: 半自動的に獲得した学習インスタンスの例

る．そして，以下に示すパターンを使い，略語が記述されている文を MEDLINE 全体から収集する．

any expression ‘(’ *abbreviation* ‘)’

ここで，*abbreviation* は作成した略語定義リストに含まれる略語である．もし，略語の括弧表現よりも前に現れる表現 *any expression* が，略語定義リストに登録されているときは，その表現を略語認識の正例とし，定義リストに登録されていない場合は，略語認識の負例とする．例えば，図 4 の上から 6 件のインスタンスは，略語 *ER* の正例であり，残りの 2 件は負例である．

今回は，30 個の略語に関して，MEDLINE におけるすべての定義とアライメントを手作業で列挙し，実験コーパスを作成した．そのうち，24 個の略語に関する事例を学習データとし，残りの 6 個の略語に関する事例を評価データとした．学習データには，24 個の略語に対応する 4,688 個の略語定義と，その事例 228,248 件が含まれる．評価データには，6 個の略語に対応する 792 個の略語定義と，その事例 89,101 件が含まれる．

CRF は C/C++言語で独自に実装した．実験には Intel Core 2 Duo 6600 (2.40GHz) 2GB メモリ のデスクトップコンピュータを用いた．生成された素性の数は 27,417 個で，対数尤度の値が上昇するスピードを見て学習を終了させた．学習に要した時間は，約 2 時間程度（2,027 イタレーション）であった．学習が比較的速く終了したのは，1 事例あたりの可能な出力ラベル系列（ラティス）の数が 5.56 (系列/事例) と少なく，forward-backward アルゴリズムのような効率計算を行わなくても，式 5 の計算に時間を要しなかったからだと考えられる．

表 3: 略語抽出システムの評価結果

システム	精度	再現率	F1 スコア
提案手法	97.2	97.9	97.5
SaRAD [1]	96.1	95.8	95.9

3.2 評価

表 3 は，ルールベースで性能が良いと言われている SaRAD [1] をベースラインシステムとして，提案手法を評価したものである．評価尺度には，精度 (precision)，再現率 (recall)，F1 スコアを用いた．提案手法は精度，再現率の両方においてベースラインシステムを上回るパフォーマンスを示し，略語定義の特徴を学習した効果が示されている．ベースラインシステムは略語と定義の語順が異なる略語定義は抽出できないが，提案手法では，*C reactive protein (PCR)* という定義が抽出できた．残念ながら，今回の評価データには並び順の異なる略語定義が 4 種類しか含まれていなかったので，提案手法がこの種の略語定義に対してロバストであるかどうか，さらに調査が必要である．

今回用いた 27,417 個の素性のうち，26,197 個（95.6%）が単語素性であった．このため，学習で得られたモデルには，“`word:0:adenosin -> O`”，“`word:0:5′-adenosine -> B1m`”のような単語に関する状態素性が大勢を占めた．今回の実験において，訓練データとテストデータでは，用いた略語が異なるため，事例中に現れる単語も全く異なる．テストデータ

の評価結果は良かったものの，特定性の高い単語による状態素性は，評価データにおいて有効に機能しないはずであり，今後は学習データの作り方を工夫するか，単語素性の絞り込みが必要であると考えている．単語素性以外の素性を調べると，“POS:+1:CC -> B1m”（次の語が等位接続詞ならば現在のラベルは B1m），“POS:0:NNS B1m”（現在の語が名詞ならばラベルは B1m）など，品詞に関する素性が上位によく表れていた．

4 結論

本稿では，略語の定義を自動認識する手法として，CRF による学習に基づく手法を提案した．ベースラインシステムと比較して，精度と再現率の両方において上回ることができた．今後は，素性の設計や学習データの作成方法を工夫するとともに，様々なコーパスで手法を評価し，提案手法のロバスト性などを検討したい．また，我々が従来提案していた統計的アプローチと組み合わせて，辞書を作るアプリケーションにおける，提案手法の効果も検証したいと考えている．

参考文献

[1] E. Adar. SaRAD: A simple and robust abbreviation dictionary. *Bioinformatics*, Vol. 20, No. 4, pp. 527–533, 2004.

[2] Sophia Ananiadou, Douglas B. Kell, and Jun ichi Tsujii. Text mining and its potential applications in systems biology. *Trends in Biotechnology*, Vol. 24, No. 12, pp. 571–579, 2006.

[3] J. T. Chang and H. Schutze. *Text Mining for Biology and Biomedicine*, chapter Abbreviations in Biomedical Text, pp. 99–119. Artech House, Inc., 2006.

[4] John N. Darroch and Douglas Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, Vol. 43, No. 5, pp. 1470–1480, 1972.

[5] Ramón A-A. Erhardt, Reinhard Schneider, and Christian Blaschke. Status of text-mining techniques applied to biomedical text. *Drug Discovery Today*, Vol. 11, No. 7–8, pp. 315–325, 2006.

[6] Toru Hisamitsu and Yoshiki Niwa. Extracting useful terms from parenthetical expression by combining simple rules and statistical measures: A comparative evaluation of bigram statistics. In Didier Bourigault, Christian Jacquemin, and Marie-C L'Homme, editors, *Recent Advances in Computational Terminology*, pp. 209–224. John Benjamins, 2001.

[7] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML-2001)*, pp. 282–289, 2001.

[8] David Nadeau and Peter D. Turney. A supervised learning approach to acronym identification. In *8th Canadian Conference on Artificial Intelligence (AI'2005) (LNAI 3501)*, p. 10 pages, 2005.

[9] Naoaki Okazaki and Sophia Ananiadou. Building an abbreviation dictionary using a term recognition approach. *Bioinformatics*, Vol. 22, No. 24, pp. 3089–3095, 2006.

[10] Youngja Park and Roy J. Byrd. Hybrid text mining for finding abbreviations and their definitions. In *2001 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 126–133, 2001.

[11] J. Pustejovsky, J. Castano, B. Cochran, M. Kotecki, and M. Morrell. Automatic extraction of acronym meaning pairs from MEDLINE databases. *MEDINFO 2001*, pp. 371–375, 2001.

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

[13] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp. 134–141, Edmonton, Canada, 2003.