大域的文脈情報を用いた英語時制誤りの検出と訂正

田尻 俊宗　　小町 守　　松本 裕治
奈良先端科学技術大学院大学 情報科学研究科
{toshikazu-t, komachi, matsu}@is.naist.jp

1 はじめに

英語学習者の数は年々増加の傾向にあり，学習者支援の需要が高まっている．[Nagata 10] によれば，語学習の学習効果を高めるためには人手の添削が非常に効果的であるが，添削には多くの時間と労力を要するため，学習者の誤りを自動で検出・訂正することで学習・教育を支援する様々な手法が提案されている．

英文の誤り検出の研究において，現在主流となっている検出対象は英語学習者が誤りやすい冠詞や前置詞などである．その一方で，冠詞，前置詞に次いで学習者に頻出する時制の誤り検出についてはこれまでほとんど研究がなされていない．その一因として，時制の決定は対象文外の情報，すなわち**大域的な文脈**に大きく依存し，そのことが時制の誤り検出・訂正を非常に困難にしていることが挙げられる [Lee 08] .

[永田 11] は，動作動詞は特殊な場合を除いて単純現在形としては用いられないという性質を利用し，単純現在形として現れる動作動詞を誤りとして検出した．しかし，この手法においては，誤り検出の対象となるのは単純現在形のみであり，誤り訂正もできない．

次の文章は日本人英語学習者コーパスである Konan-JIEM Learner Corpus Second Edition（KJ コーパス）[1] で見られる事例である[2].

(1) I had a good time this Summer Vacation.
First, I *go/went to KAIYUKAN with my friends.

この例では 2 文目の go を went とするのが正しく，検出だけであれば永田ら (2011) の手法を用いれば可能であるが，訂正に関しては，未来形とするか，過去形とするかは大域的な文脈に依存する[3]．この事例の場合，前文において this Summer Vacation が過去の事象として参照され，対象文で First を用いていることから前文を参照していることが分かれば，過去形が正しいということが分かる．少なくとも人間はこのような推論によって誤りの訂正を行うが，これを機械的に訂正することは容易なことでない．一つの方法として，大域的な文脈を素性として組み込み，機械学習を用いて検出・訂正を行うことが考えられる．本論文では，大域的な文脈を自然に捉えるため，時制の誤り訂正を文章中の各動詞句に正しい時制ラベルを振る系列ラベリング問題と捉え，前の動詞句との組合せ素性を考慮した．その結果，時制誤り検出・訂正の精度の向上に貢献したことを示す．

2 時制誤りコーパスの作成

高精度の時制誤り検出・訂正システムの構築には，時制誤りタグの付いたコーパスが必要となる．しかし，既存の時制誤りタグ付きコーパスは，規模が充分でないなどの理由で学習に適していない．そこで，我々は外国語学習 SNS である「Lang-8（ランゲート）[4]」に着目した．

Lang-8 では，世界 180ヶ国から 30 万人もの人々が参加し，学習したい言語でエントリ（日記）を書き，お互いに添削しあうことで語学力を高めることができる．総エントリ数は 120 万にのぼり[5]，非常に大規模で有用な言語資源と言える．我々は，この Lang-8 から大規模な時制誤りコーパスを作成し，時制誤り検出・訂正システムの構築に用いた．

2.1 時制誤りコーパスの作成手順

Lang-8 からの時制誤りコーパスの作成手順について述べる．

1. Web クローリングを行い，Lang-8 の HTML データを収集する[6]．その結果，622,233 のエントリを収集した．
2. そのデータの中から英語で書かれた 180,209 のエントリを抽出する．この際，英語以外の文字が用いられているエントリと全く添削がされていないエントリは除き，126,482 エントリとなった．
3. ノイズとなりうる添削文を除去する．添削文中にコメントが入っているものや，添削前の文と大きく変化している添削文を除き，876,223 の添削文を 752,701 文に絞り込んだ．
4. 添削前と添削後の文について，動詞句の時制を同定した上でマッチングを行い，各動詞句の時制が異なれば時制誤りとする．

[1] http://gsk.or.jp/catalog/GSK2011-B/catalog.html
[2] 「*a/b」は，a が誤用例を，b が訂正例を表す
[3] なお，本論文では，テンス（過去，現在，未来）とアスペクト（基本，完了，進行，完了進行）を組み合わせた 12 種類の文法形式を時制と定義する．
[4] http://lang-8.com/
[5] 2012 年 1 月現在
[6] 2011 年 10 月 7 日までのデータ

表 1: CRF の訓練に用いた各動詞句に対する素性

略称	説明
j-learn	学習者の書いた時制（表層の時制）
head	見出し語
L	左の語
R	右の語
nsubj	主語
dobj	目的語
aux	助動詞
prep	前置詞で繋がる語
xcomp	並列されている動詞句の見出し語
p-tmod	時間副詞句
time	正規化時間副詞
advmod	その他の副詞
main-sub	従属接続詞（対象動詞句が従属節内のとき）
sub-main	従属接続詞（対象動詞句が主節内のとき）

以上の手順に沿って，126,482 エントリ，2,112,288 動詞句のコーパスを作成した．

3 文脈情報を用いた時制誤り検出・訂正の手法

1 節でも述べた通り，時制誤りの検出・訂正を行うには，対象の動詞句に関する情報を用いるだけでは不十分であり，対象の動詞句以外の情報，すなわち大域的な文脈情報を用いて，前後の語句との関係を考慮しなければならない．よって，検出・訂正システムを構築する際には，何らかの形で大域的文脈の情報を組み込む必要がある．本論文では大域的文脈情報を考慮するため，時制誤りの検出を文章中の各動詞句に正しい時制ラベルを割りあてる系列ラベリング問題と捉える．系列ラベリングには，ラベル推定の精度が高く，柔軟な素性を組み込むことができる条件付確率場 (CRF) [Lafferty 01] を用いる．

3.1 訓練に用いる素性の選択

訓練に用いる素性には表 1 のものを用いる．

素性の生成には，Stanford Parser 1.6.9[7]を用いた．nsubj, dobj, aux, prep, xcomp, advmod については，Stanford Parser の Typed Dependency をそのまま用いた．main-sub, sub-main については，対象の動詞句を含む文が複文であるとき，対象の動詞句が従属節内にあり，かつ文が主節，従属節の順となっている場合に main-sub の値を従属接続詞にする．また，対象の動詞句が主節内にあり，かつ文が従属節，主節の順となっている場合に sub-main の値を従属接続詞にする．

(2) It pours when it rains.

(3) When it rains it pours.

例えば，文 (2) では，動詞句 rains の main-sub 素性を when にする．また，文 (3) では，動詞句 pours の sub-main 素性を when にする．

p-tmod については，Typed Dependency の tmod（時間副詞）がついた語を句単位で抽出する．

[7]http://nlp.stanford.edu/software/lex-parser.shtml

表 2: 素性 time の各値と，対応するキーワード

値	キーワード
過去	yesterday または last
現在	now
未来	tomorrow または next
THIS	today または this

表 3: 素性テンプレート

局所的素性
<head> <head, j-learn> <head, L，R> <L> <L，head> <L，j-learn> <R> <R，head> <R，j-learn> <nsubj> <nsubj, j-learn> <aux> <aux, head> <aux, j-learn> <prep> <prep, j-learn> <xcomp> <xcomp, head> <xcomp, j-learn> <time> <time, j-learn> <advmod> <advmod, j-learn> <tmod> <tmod, j-learn> <main-sub> <main-sub, j-learn> <sub-main> <sub-main, j-learn>
大域的文脈素性
<p_tmod′> <p_tmod′，j-learn> <p_tmod′，j-learn′> <p_tmod′，j-learn′，j-learn> <time′> <time′，j-learn> <time′，j-learn′> <time′，j-learn′，j-learn>

(4) I had a good time last night.

例えば，文 (4) では，動詞句 had に対して night が tmod として与えられる．このとき，night は名詞句 last night の一部であるので，動詞句 had の p-tmod として last night を与える．

また，time は p_tmod を「過去」「現在」「未来」「THIS」のいずれかに正規化したものである．例えば，文 (4) では，動詞句 had の p-tmod として last night が与えられており，これを過去を示す時間副詞と同定し，time として「過去」が与えられる．

time の各値と，対応するキーワードを表 2 に示す．p-tmod に対応するキーワードが含まれる場合，正規化が行われる．なお，THIS は，時間副詞のうち，幅を持つものに付与される．

3.2 素性テンプレートの作成

素性同士の組み合わせを表した素性テンプレートを表 3 に示す．項目テンプレート内の<a, b>は，素性 a と素性 b を組み合わせた素性関数を用いることを表している．a′ は，対象動詞句の 1 つ前の動詞句の素性 a を表す．局所的素性のテンプレートは，対象動詞句内での素性を組合せた素性関数であり，大域的文脈素性のテンプレートは，前の動詞句の素性との組合せを含む素性関数である．

(5) I **went** to Kyoto yesterday.
I ***eat** yatsuhashi and **drank** green tea.

例えば，文章 (5) において，eat の 1 つ前の動詞句は went であり，drank の 1 つ前の動詞句は eat である．

表 3 の素性テンプレートを用いて，3.1 節の素性から素性関数を生成し，モデルの学習に用いる．

3.3 機械学習による時制誤り検出・訂正

以上の手順で学習したモデルを用いて，正しい時制が未知である動詞句に対して，正しい時制を推定すること

ができる．このとき，対象の動詞句に対してどのように時制誤りの検出・訂正を行うかを考える．

最も単純な方法としては，対象の動詞句の表層から同定した時制と，モデルの推定した時制を比較する方法が考えられる．このとき，両者が同じ場合は正しい時制であるとみなし，異なる場合は誤りとして検出する．さらに，誤っている場合には，モデルの推定した時制を訂正候補として提示する．

しかし，表層の時制を素性として組み込んだ場合にこの方法で検出・訂正を行うと，1 つの問題が生じる．表層の時制を素性として組み込むと，検出性能において，Precision の向上が期待できる代わりに，Recall が低下する．これは，ほとんどの事例において表層の時制と正解の時制が一致することに起因している．もちろん，時制誤り検出システムにおいては高い Precision を出せることが望ましいが，あまりに Recall が低い場合は，検出自体を行わなくなり，システムが成り立たない．よって，実用的なシステムの構築を最終的な目標に置くのであれば，Precision と Recall のトレードオフの関係を導く必要がある．そこで，CRF において，ラベルの推定に用いる各ラベルの周辺確率を用いる．

CRF では，周辺確率が最も高いラベルを対象要素の局所的な推定ラベルとする．そして，系列全体の推定ラベルの組合せのうち，最も尤度の高い出力ラベル列を最終的な推定結果としている．推定ラベルの周辺確率は，そのラベル推定の確信度とみなすことができる．この確信度が低い場合は推定結果を信じないとすれば，Precision と Recall を調節することができる．以下でその方法を具体的に述べる．

まず，対象の動詞句の表層時制と，モデルの推定した時制を比較する．両者が異なる場合は，最も単純な検出方法と同様に，誤った時制であるとみなし，推定結果を訂正候補として提示する．両者の時制が同じ場合，推定ラベルの周辺確率と，あらかじめ定めた閾値を比較する．周辺確率が閾値以上であれば，ラベル推定の確信度が高いとみなせるので，推定通り正しい時制であると判定する．周辺確率が閾値を下回る場合，ラベル推定の確信度が低いとみなせるので，モデルの推定を信じず，時制誤りとして検出する．このとき，訂正候補として提示するのは，モデルの推定ラベル以外で最も周辺確率の高いラベルである．この時制誤り検出・訂正手法を **T-CRF** と表記する．この手法を用いれば，閾値を上げれば上げるほど Precision が低くなる代わりに Recall が高くなるというトレードオフの関係を導くことができる．

4 時制誤り検出性能の評価実験

評価実験を行うにあたって，2 節で作成した Lang-8 の時制誤りコーパスを学習に 10 万エントリ，テストに 1,000 エントリをそれぞれ用いる．テストデータは，16,308 の動詞句を含み，そのうち 1,072 の動詞句 (6.6%) が時制誤りを含む．使用ツールは，素性生成や時制の付与に Stanford Parser 1.6.9 を用いた．また，ベースラインで用いるサポートベクトルマシン (SVM) の学習には LIBLINEAR 1.8 [8] を，CRF の学習に CRF++ 0.54 [9] を用いた．LIBLINEAR と CRF++のパラメータは，それぞれデフォルト値を用いた．

4.1 ベースライン

CRF によって大域的文脈情報を用いる効果を調べるため，比較対象として次の 2 つの手法をベースラインとする．

- **SVM** : 局所的素性のみを用いて多クラス SVM を学習する
- **G-SVM** : 局所的素性と大域的文脈素性を用いて多クラス SVM を学習する

4.2 提案手法

- **U-CRF** : 直前の動詞句の推定ラベルを考慮せず，局所的素性と大域的文脈素性を用いる
- **B-CRF** : 直前の動詞句の推定ラベルを考慮して，局所的素性と大域的文脈素性を用いる

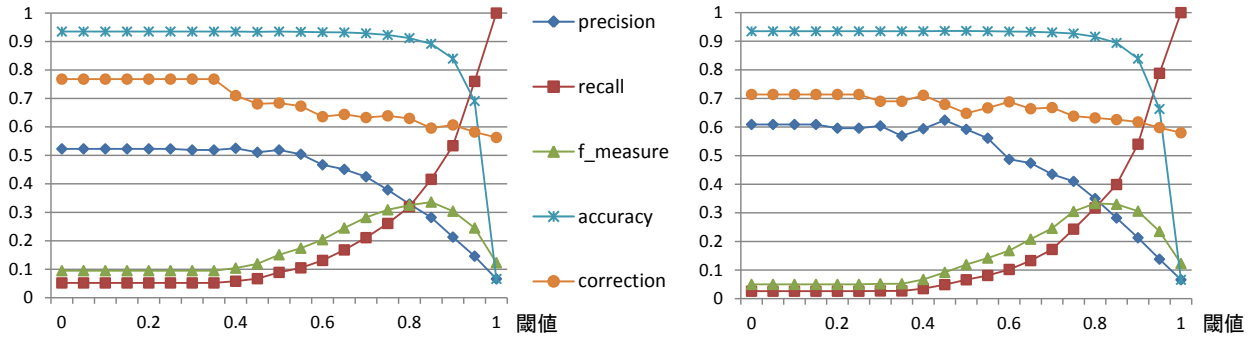
検出・訂正の基準は，3 節で述べた **T-CRF** を用いる．

4.3 実験結果

それぞれの手法の検出･訂正性能を表 4 に示す．**検出性能**は，Precision, Recall, F 値，Accuracy によって評価する．さらに，正しく時制の誤りを検出できた事例のうち，訂正候補も正しく提示できた事例の割合を Correction によって表し，これを**訂正性能**の指標とする．**U-CRF** と **B-CRF** に関しては，F 値と Accuracy がそれぞれ最大となる場合の値を示している．また，図 1(a) に **U-CRF** の性能を，図 1(c) に **B-CRF** の性能を示す．ここで，横軸は CRF の推測結果に対して足切りを行う閾値を表し，縦軸は閾値を変えたときの各指標の値を表している．

表 4 の **SVM** と **G-SVM** を比較すると，全ての値において **G-SVM** が上回っており，大域的文脈素性を用いる効果があることが分かる．また，CRF を用いた手法では，**T-CRF** の閾値によって性能が変動するが，全体的に **G-SVM** よりも良い性能を示している．よって CRF によって大域的文脈素性を用いる効果があることが分かる．**B-CRF** と **U-CRF** を比較すると，Precision は **B-CRF** の方が高く，Recall は **U-CRF** の方が高い傾向にある．また，Accuracy は **B-CRF** の方が高く，Correction は **U-CRF** の方が高い．

[8] http://www.csie.ntu.edu.tw/~cjlin/liblinear/
[9] http://crfpp.sourceforge.net/



(a) **U-CRF** (c) **B-CRF**

図 1: **U-CRF**, **B-CRF** の閾値ごとの検出・訂正性能

表 4: 各手法の時制誤り検出・訂正性能

手法	P	R	F	Acc	Corr
SVM	.293	.089	.136	.926	.537
G-SVM	.301	.090	.139	.926	.546
U-CRF（0.5）	.519	.089	.151	.935	**.684**
U-CRF（0.85）	.282	**.416**	**.336**	.892	.596
B-CRF（0.5）	**.592**	.066	.119	**.936**	.648
B-CRF（0.8）	.350	.317	.333	.916	.632

5 考察

B-CRF の閾値を 0.8 にしたときの実験結果を基に，考察を行う．時制ごとの検出性能を見ると，最も頻度の高い単純現在形から単純過去形への誤りは，211 個のうち，61 個が正しく検出，52 個が正しく訂正できていたが，次に頻度の高い単純過去形から単純現在への誤りは，94 個のうち検出・訂正ともに 9 個のみであった．この性能差の一つの要因は，[永田 11] が示すように，動作動詞が単純現在形として用いられている場合は，誤りとみなせるものがほとんどであるので，時制誤り検出が比較的容易であることが挙げられる．

次の文章は，単純過去形から単純現在形への誤りのうち，検出ができなかった例である．

(6) 添削前：The place I ***lived** is a warm city.
添削後：The city I **live** in has a warm climate.

この例においては，lived と is で時制が異なることが分かれば，どちらかが誤っていることが分かる．しかし，今回は大域的文脈素性を前の動詞句との組合せのみに限定しており，is を過去形であると判定することはあっても，lived を現在形であると判定することはできず，この例を正しく検出することができない．今後，後の動詞句との組合せを考慮することで，より大域的文脈を考慮できることが期待される．

もう一つの問題点としては，作成したコーパスが挙げられる．今回作成したコーパスにおいては，添削のつかなかった動詞句については，全て正しい時制であるとみなしている．しかし，全ての添削者がエントリ全体を見て添削しているとは限らない．よって，コーパス内に本当は時制が誤っているが，添削されずに正しい時制であるとみなされたものが含まれ，学習に悪影響を与えている可能性がある．

6 おわりに

本論文では，大域的文脈を用いて英語の時制誤りの検出と訂正を行う手法を提案した．システムの構築のため，大規模な時制誤りコーパスを作成し，CRF での大域的文脈情報を用いた学習を行った．評価実験の結果，大域的文脈素性が時制誤りの検出・訂正の精度の向上に貢献していることを示した．

今後の課題としては，大域的文脈情報をより自然に組み込めるモデルの考案，素性の洗練などが挙げられる．また，今回の評価実験においては，Lang-8 から作成したコーパスを用いて性能を評価したが，考察で述べたように，Lang-8 から作成したコーパスに付与された時制の信頼度は決して高いとは言えない．よって，KJ コーパスなどの人手で作成したコーパスでの評価実験も今後の課題としたい．

参考文献

[Lafferty 01] Lafferty, J.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in *Proceedings of ICML*, pp. 282–289 (2001)

[Lee 08] Lee, J. and Seneff, S.: Correcting misuse of verb forms, in *Proceedings of the 46th ACL*, pp. 174–182 (2008)

[Nagata 10] Nagata, R. and Nakatani, K.: Evaluating performance of grammatical error detection to maximize learning effect, in *Proceedings of COLING*, pp. 894–900 (2010)

[永田 11] 永田亮, Sheinman, V.：Stativity 判定に基づいた時制誤り検出, 言語処理学会第 17 回年次大会 発表論文集, pp. 1055–1058 (2011)