

外国語作文における自動添削手法について

西村 則久[†], 安村 通晃

慶應義塾大学大学院 政策・メディア研究科
〒252-8520 神奈川県藤沢市遠藤 5322
[†]bud@mag.keio.ac.jp

Abstract

外国語作文の自動添削では、通常の CAI ソフトウェアは数例の正解のみを内部に持っている場合が多いが、この場合、非常に簡単な問題を除いては多様な解答例に対応することができない。

本報告で提案する外国語作文の自動添削手法では、正解の一括表現を用いて多様な解答例に対応することが可能である。このシステムを試作し、評価を行った。本発表ではシステムの仕組み、評価結果について述べると同時に正解一括表現法として BUD(Basic Universal Description) を提案する。

A Method of Automatic Correction of Foreign Language Composition

Norihisa Nishimura, Michiaki Yasumura

Graduate School of Media and Governance, Keio University
5322 Endo, Fujisawa, Kanagawa, 252-8520 JAPAN

Abstract

Concerning the automatic correction of foreign language composition exercises, conventional software on the market contains only a few correct answers. This approach, however, cannot handle a lot of various correct answers, except when the question is simple.

We propose a method of automatic correction of foreign language composition exercises which deals with an enormous number of correct answers, making use of multiple-unit representation which expresses multiple sentences all together. We built a prototype system and made some evaluations. Here we describe the mechanism of the system and the result of the evaluations. We propose BUD(Basic Universal Description) as a multiple-unit notation.

1 はじめに

近年のコンピュータの教育現場における普及には目覚ましいものがあり、この数年間で学校でのコンピュータの導入率・導入台数は著しく増加し、特別教室では一人一台という環境が現実のものとなってきた。これに伴い、授業時間内外を問わずコンピュータの教育利用に関する研究や実験的運用も盛んになり、新たなメディアを利用したCALL(Computer Assisted Language Learning)への期待が非常に高まっている。

しかしながらCALLの現時点での問題点の一つとして、ユーザ・インタフェースや入出力手段は日々発展しているものの、入力に対する適切な評価(採点や指示)を行う方法が確立されていないことが挙げられる。すなわち、学習者は出題された問題を単に解くだけであり、ソフトウェアに本来期待されるべき誤りに対する丁寧な対応を得ることができていない。

これを外国語学習の読む、書く、聴く、話すという四技能のうち、書く能力のためのソフトウェアについて考えてみる。ほとんどのソフトウェアでは出題の形式は並べ替え問題、穴埋め問題、選択問題であり、学習者に自由に書かせる作文問題は少ない。これは自由記述形式の作文問題では学習者からの様々な入力に対する適切な評価を下すことが困難だからである。事実、作文問題を出題している数少ないソフトウェアでは、学習者の解答があらかじめ用意された数例の正解のうちのどれかと一致すれば正解、一致しなければ不正解としているが、それだけでは十分適切な指示とは言えず、学習者の本来の期待に添っているとは言い難い。自然言語処理や機械翻訳の手法を採り入れた人工知能的アプローチは非常に期待されるが、少なくとも現時点では実用化までには至っていない。

しかし、書く能力に関する限り作文問題は最も真の力を問うものと考えられており、作文問題で適切な自動添削の手法を開発できれば、その有用性は大きい。

本研究で提案するシステムは、英作文問題において上述した問題の解決を図るもので、ここにその実現方法を述べる。

2 英作文添削システム

我々は自由記述に対応する英作文添削システムを開発した。本システムは、学習者の解答に赤入れして正解を提示する部分と学習者の解答の誤りに対してコメントを表示する部分とで構成されている。

2.1 正解処理

英作文問題では正解とされる答えが通常複数存在する。なおかつその数は膨大になる場合が多い。本システムは膨大な数の正解のすべてと学習者の解答との比較を行い、学習者の解答をそれに最も近い正解になるように赤入れを行う。

2.1.1 正解一括表現法 BUD

膨大に存在する正解のデータをそのまま人間の手で一例ずつ作成するのは労力も大変でありデータ量も膨大になるため、非常に簡単な問題の場合を除き実現困難である。本研究では複数の正解を少ないデータ量とわずかの労力で実現するための一括表現を考案した。図1に正解一括表現法BUD(Basic Universal Description)の例を示す。この表記では省略可能を表す記号()と交換可能を表す記号[]を用いる。正解を普通に記述した上で省略可能な箇所は()に入れる。図1の例ではchildが単数型childrenが複数型の場合に正解という意味でchild(ren)と表現する。同様にwood(s)と表現する。他の語句と交換可能な箇所では[]の中で交換可能なすべての語句をカンマで区切って並べる。この例では「森」にあたる表現がforestとwood(s)のどちらでも正解という意味で[forest, wood(s)]と表現している。このように、括弧の中に更に括弧を記述した多重括弧も可能である。この一括表現は六通りの正解を表している。

he took his child(ren) to the [forest, wood(s)].

↓
He took his children to the forest.
He took his child to the forest.
He took his children to the woods.
He took his child to the woods.
He took his children to the wood.
He took his child to the wood.

図1: 一括表現の例(1)

図2に示すように複数の一括表現を正解データとして登録できるようになっており、文型の全く異なる別解が存在する場合は、文型毎に一括表現で記述する。これにより、正解データが十分なものであれば、学習者の解答がどの文型を採用していてもそれに近い正解が用意されていることになる。

```
[we, you] can hardly forget infant memories.
it is [difficult, not easy] to forget infant
memories.
infant memories [last long, die hard].
↓
We can hardly forget infant memories.
You can hardly forget infant memories.
It is difficult to forget infant memories.
It is not easy to forget infant memories.
Infant memories last long.
Infant memories die hard.
```

図 2: 一括表現の例 (2)

前述した二種類の括弧を用いた記号を用いてかなり多くの文が比較的小規模なデータで表現できるが、正解とすべきでない文が含まれてしまう可能性がある。図3では、“by the homer by Oh”という表現を正解に含めない為に二文に分割して記述しなければならない。この例では僅か二つの文で全てを含むことができるが、実際に膨大な正解を記述する場合には文の分割が数多く発生して結局データ量・作成にかかる労力共に膨大なものになってしまう。このような文の分割を避けるために排他関係を表す記号‘*’を導入した。図4のように、記号‘*’が添えられた語句同士のパターンは排他的であり、同時には正解に含まれないことを表している。この記号は直後に識別子を付けることも可能であり一つの一括表現文内に識別子の数だけ排他関係を指定できる。

```
with the home run [by, of] Oh, ...
by the home run of Oh, ...
↓
with the home run by Oh, ...
with the home run of Oh, ...
by the home run of Oh, ...
by the home run of Oh, ...
(by the home run by Oh, ... を除外する)
```

図 3: 分割された表現

他にも文の分割を避けるための記号を用意した。異なる位置に存在できる語句を{ }を用いて一つの一括表現文で表すことができる(図5)。呼応する語句があるケースも< >を用いて一つの一括表現文

```
[with, *by] the home run [*by, of] Oh, ...
↓
with the home run by Oh, ...
with the home run of Oh, ...
by the home run of Oh, ...
```

図 4: 排除される組み合わせ

で表すことができる(図6)。すなわち、< >内の各順番に現れたものは、その順で用いなければならない。これらの記号も{及び<の直後に識別子を付けることで識別子の数だけ用い分けられる。

```
{also,} you can {also} take out the foods
{, also}.
↓
Also, you can take out the foods.
You can also take out the foods.
You can take out the foods, also.
```

図 5: 文中の異なる位置に存在できる語句

```
<we, you> must keep <our, your> promises.
↓
We must keep our promises.
You must keep your promises.
```

図 6: 呼応する語句

C言語のdefineと同様のマクロ定義記号も用意した(図7)。これには#preと#postがある。#preは一括表現文の展開前に置換処理が行われるもので、一括表現文が複数ある場合に共通の部分を見易くするものである。#postは一括表現の展開後に交換処理が行われるもので、交換前の表現も正解として残す。これは主として省略形などに利用する。

2.1.2 近似度計算方法

正解データに含まれる全ての正解例のうち、学習者の解答に最も近いものを求めるために、学習者の解答と正解例の近似度を全ての正解例について求める必要がある。

ある一つの前解と学習者の解答との近似度の求め方としては、二つの文で一致する単語の間に交差せずに引くことができる線の最大本数を求め、これが大きいほど近似度が高いとした(図8)。この例では、正解2が正解1よりも学習者の解答に近いと判

```
#pre "DEED" "they dislike him"
#pre "REASON" "because he is not polite"
#post "he is" "he's"
#post "is not" "isn't"
```

REASON, DEED.
DEED REASON.

↓
Because he is not polite, they dislike him.
Because he's not polite, they dislike him.
Because he isn't polite, they dislike him.
They dislike him because he is not polite.
They dislike him because he's not polite.
They dislike him because he isn't polite.

図 7: 定義記号

断される。線の本数が同じになる正解が複数存在する場合は、線の引かれていない単語の数が少ない例を近似とした。更にこの数も同じ正解が複数存在する場合は、先に求められた正解を優先している。このため、正解データを作成する際には、標準的と考えられる文型や語句から先に記述する必要がある。

```
解答 Please see off my father. 3本
正解1 Will you see my father off?
解答 Please see off my father. 4本
正解2 Will you please see my father off?
正解3 .
正解4 .
正解5 .
```

図 8: 最も近い正解の探索法の例

2.1.3 正解探索処理の高速化手法

学習者の解答に最も近い正解を探索する処理を高速化するために、学習者の解答と正解の一括表現の中にある語を短い中間コード、すなわち、文字列に対してユニークな数を割り当てて置き換えることにした。この高速化により、処理時間が35～40%短縮した。

2.1.4 赤入れ添削による表示方法

学習者の解答とそれに最も近い正解とを比較し、学習者の解答の中で不要な語句は削除、不足してい

る語句は挿入、それらが同じ箇所が存在する場合は置換を示す赤入れ処理を行う(図9)。

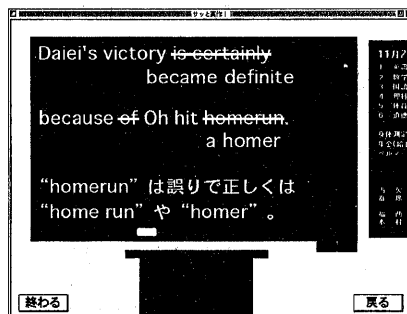


図 9: 添削画面

2.2 誤答処理

学習者の解答の中に誤りと判断される語句が含まれていた場合、適切なコメントを学習者に返すことが重要である。このため、システムは用意された文字列を表示する。例えば、「約束は守りましょう。」を英作文する際に学習者が“protect”という単語を用いると、システムは「“protect”は人や物を危険などから守ることなので適当ではありません。」という文字列を表示する。これを可能にするために各問題は例えば図10のような誤答データを持つ。

```
protect
人や物を危険などから守ることなので適当ではありません。
defend
敵・危害などから守ることなので適当ではありません。
```

図 10: 誤答データの例

3 評価と考察

3.1 アンケートによる評価

評価用問題データを13問作成し、学生9人と英語教員3人の協力を得て評価を行った。学生には実際に全問題の添削を受けさせ、アンケートを行った。教員には学生が受けた添削の内容を提示して、アンケートを行った。

学生のアンケート結果（五段階評価）を図 11 に示す。結果は概ね良好で、特に本システムが「英作文能力向上への効果があるか？」の問いに対しては平均 4.3 の回答が得られた。しかし「添削は人間の教師のレベルか？」の回答は他と比較して特に肯定的ではなかった。自由記入欄で得られた意見としては、「全くわからない人のためにヒントが見られると良い」「添削時に解説が不足している」「受験英語の学習法としては良いと思う」「模範解答として複数の正解を挙げるともっと良い」などがあった。図 12 に実際に学生の受けた添削結果の一部を示す。

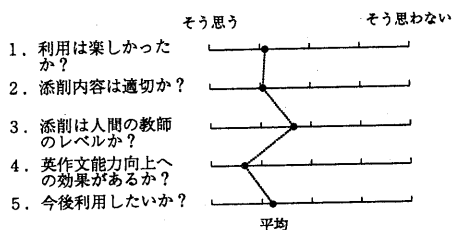


図 11: 学習者アンケート結果

教員にも同様のアンケートを行った。肯定的な教員がいる一方で、選択項目は「一概には答えられない」とされるものも多かった。自由記入欄では、提示された正解に誤りがある点を指摘し、正解データの信頼性を高めれば学習効果を上げることができるという意見があった。

3.2 正解データ作成について

少量のデータから効率的に正解を生成することができた。実験で使用した正解データの行数および正解数を図 13 に示す。平均では約 18 行の正解データで 50,000 通り以上の正解を表現した。

正解データは念入りに作成しても抜けが生じ、予期していなかった正解が存在するのが常となるから、それを含むようにデータを随時修正することになる。こうして正解データの完成度が高められるまでにはある程度の期間が必要である。

「抜け」以外に、誤った文を正解に含まないように注意が必要である。データは予め言語の専門家やネイティブスピーカーによる十分なチェックを受けるのが望ましい。

問題「君は早くも良い案を思いついたようだね。」

You hit

大きく外れています。

I ~~think~~ you ~~may~~ have a good idea ~~early~~.
guess already

You already have a ~~great~~ idea.
seem to good

I ~~know~~ that you ~~get~~ a good idea.
guess already have

You already seem to ~~hit~~ on a good idea.
have a

You have a good idea, don't you?
already

You have a good idea, don't you?
already

You already have a good idea, don't you?

正解です。

You ~~look hitting~~ on a good idea ~~early~~.
must have already hit
a

図 12: 実験での添削内容（例）

4 今後の課題

現在、プロトタイプシステムは開発して間もないため、次に示すような課題があり、今後、これらについて取り組んでいく予定である。

4.1 正解探索処理の高速化

正解データで表現される正解の数が非常に多い場合に探索に時間がかかるため、正解探索処理の高速化が課題の一つである。

問題	行数	正解数
彼女はクラスの中で誰よりも背が高い。	7	636
約束は守りましょう。	13	12751
その店は10時まで開いていた。	1	56
君は早くも良い案を思いついたようだね。	6	203
彼はたぶん6時までには家に帰ってくるだろう。	6	1152
彼は自分勝手だったので客に好かれなかった。	29	4696
人は時には得意でないことをやらなければならない。	4	11928
私は私の幸運を当然と思っている。	4	192
海外旅行をするには、パスポートを取得しておかなければならない。	8	195200
英語を話せることと国際人であることは違う。	5	4310
小さい頃に覚えたことは、なかなか忘れないものだ。	19	248024
それが見るに値するか否かに関わらず、私はそこへ行きたくない。	6	6080
王がホームランを打ったのでダイエーの勝利が決定的になった。	52	23122
平均	17.8	56483

図 13: 正解データ行数・正解数

4.2 幅広い誤りへの対応

現在行っている誤答処理は単純なので認識できる誤りの形態に限られている。スペルチェックの併用など、本方式ではカバーしていない事例に対応するような誤答処理の強化も必要である。

4.3 正解データ作成支援ツール

一括表現を用いた正解データをより簡単に作成できるよう、GUIを用いたエディタなどの教材作成ツールの開発が望まれる。

5 まとめ

外国語作文自動添削手法および正解の一括表現法 BUD(Basic Universal Description) を提案し、プロトタイプの実装した。現在、英語版「サッと英作!」を開発し、中国語版「サッと中作!」を開発中である。どちらも Macintosh 版と Windows 版がある。これらは近日中にインターネットで一般

に公開する予定である。

参考文献

- [1] 今田敬, 小西達裕, 高木朗, 小原啓義, 機械翻訳の技術を用いた英語教育用知的 CAI システム, 1991 年度人工知能学会全国大会 (第 5 回) .
- [2] 菅井勝雄, 野嶋栄一郎, 知的教育システムと学習, 共立出版, 1992.
- [3] 北尾謙治監修, 野澤和典, 島谷浩, 山本雅代, コンピュータ利用の外国語教育, 英潮社, 1993.