

構文解析を用いた文章推敲支援システム

2S-9

杉江美佐子 *後藤 浩文 中里茂美 大黒和夫

(株)東芝 情報通信システム技術研究所 * (株)東芝ソフトウェアエンジニアリング

1. はじめに

我々は従来より、文章の推敲を目的とするシステムの開発を行ってきた。ここでは、より豊かな表現にするための類語変換、文体の統一、文法的な誤りの指摘等を実現し、商品化した。

今回、従来の技術を一步進め、語と語の関係も含めた推敲を行えるシステムの試作を行った。

本報告では、このようなシステムの構成とその機能について述べる。

2. システムの構成

システムは、従来の推敲支援システムに構文解析、構文生成、および、構文変換を新しく加え、また、語の関係をチェックする規則を新たに持った構成になっている。

本システムの構成を図1に示す。網かけされている部分が、今回新たに加えた部分である。

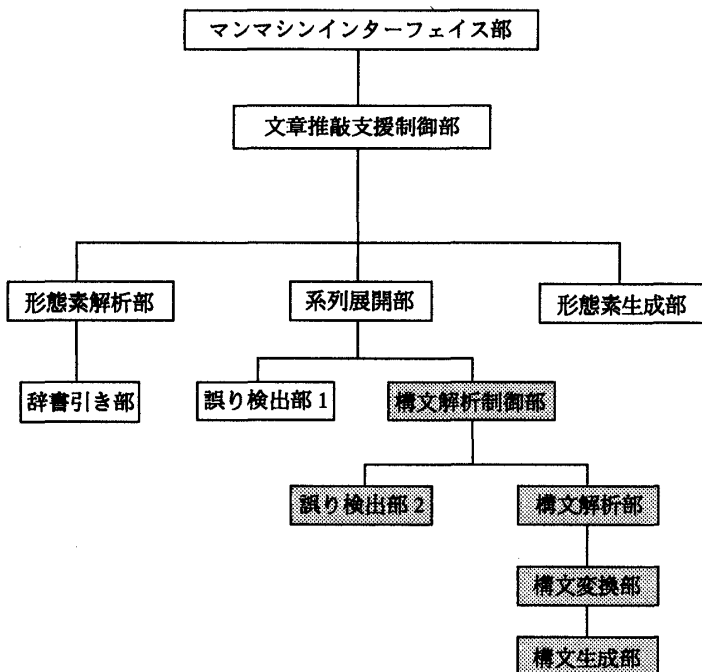


図1 システムの構成

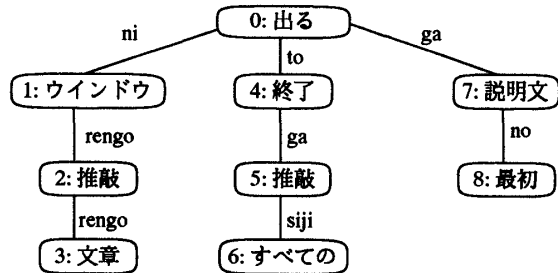


図2 (文1)の解析結果

3. システムの機能

文体の統一など、形態素の持つ情報を操作することによって行える文章の推敲は誤り検出部1で行っている。構文解析の結果を用いた文章の推敲は、誤り検出部2で行っている。

誤り検出部は、検出規則と処理プログラムよりなる。検出規則は検出する不具合の種類ごとに記述されていて、条件とアクションからなっている。検出規則は、プログラムの変更なく書き換える事が出来るため、規則の修正は容易に出来る。

次に、誤り検出部2で行っている処理の中から下記のものゝ具体的な例をあげて説明する。

- a. 語の順番誤りの検出
- b. 意味的な誤りの検出
- c. 対になる語の誤りの検出

(語の順番誤りの検出)

(文1) 最初の説明文が、すべての推敲が終了すると、文章推敲ウィンドウに出ます。

(文1)は、図2のように解析される。この図で、数字で示されているのは、元の文での語の順番である。番号は文末から振られている。主語と述語は近いほうが分かりやすいため、「語の順番誤り検出規則」では図3のような規則が記述されている。この規則によって、トップ・ノードから"ga"アークで繋がっている「説明文」の順番と、

```

if (0(ga_2 to_1))
then {
  if (1:1. 順番 < 2:2. 順番)
  then {
    警告を表示
    1:1 を先に生成する
  }
}

```

図3 語の順番誤り検出規則の一部

"to" アークで繋がっている「終了する」の順番が比べられ、この場合は「終了する」の方が小さいのでチェックに掛かり、警告が出される。また、構文生成で生成される順番が変更され変換候補として次の文が示される。

(文2) すべての推敲が終了すると、最初の説明文が、文章推敲ウィンドウに出ます。

(意味的な誤りの検出)

(文3) 私は友人に偶然合った。

(文3)の解析結果は図4のようになる。誤り検出部2の「意味的な誤り検出規則」には、図5の様な規則が書いてある。また、図4の「合う」は、図6(a)の様な格パターンを持つ。「私」は人であるため、「合う」の格パターンとはマッチしないので、各々の同音語で共起するものが調べられる。この場合は、「あう」の同音語である「合う」と「会う」で格パターンとのマッチングが行われる。図6(a),(b)より、図7の様な結果になる。従って、「合う」に警告が与えられ、変換候補として、「会う」が示される。

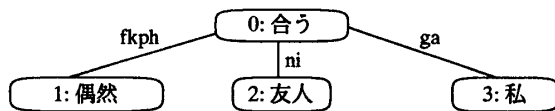


図4 (文3)の解析結果

```

if (格パターンとマッチしない)
then {
  if (同音語候補がある)
  then {
    if (各同音語に格パターンとマッチするものがある)
    then {
      警告を表示
      候補を表示
    }
  }
}

```

図5 意味的な誤り検出規則の一部

```

(趣味・意見) ga (人) ni 合う
(洋服) ga (人) ni 合う

```

図6(a) 「合う」の格パターン

```

(人) ga (人) ni 会う

```

図6(b) 「会う」の格パターン

動詞	ga (私)	ni (友人)	結果
合う	×	○	×
会う	○	○	○

図7 格パターンのマッチング結果

(対になる語の誤りのチェック)

(文4) その話は全然面白い。

(文4)の解析結果は図8のようになる。言葉の中には、それ以降に来る言葉の種類を限定するものがあるため、「対になる語の誤り検出規則」には図9の様な規則が書いてある。(文4)で副詞は「全然」であり、「全然」の上方にある「面白い」は、否定ではないため、「全然」に警告が与えられ、変換候補として「とても」が示される。



図8 (文4)の解析結果

```

if (副詞)
then {
  if (ソース・ワード「全然」の上方が否定ではない)
  then {
    警告を表示
    候補として(副詞:とても)を表示
  }
}

```

図9 対になる語の誤り検出規則の一部

4. おわりに

構文解析を用いることによって、より高度な推敲を行えるようになった。しかし、処理の単位が一文のため、文に渡った解析を行えない。また、文章の推敲をするためには、文章の内容をよく理解して行う必要があるが、本システムでは文章の内容を把握するまでにはなっていない。将来的には文章全体を処理の単位とし、また、意味解析を行えるシステムにしていく予定である。

参考文献

小山、小林、中里、斎藤、:"文章作成支援システムの機能について"、昭和62年第35回情報処理全国大会 4S-4