日本語文書校正支援ツールの開発
－動詞格フレームと名詞シソーラスの利用－

1W－8

納富 一宏　増田 進二　加藤 達矢　大野 聡　内山 明彦
早稲田大学理工学部

1．はじめに

従来の日本語文書を対象とした校正・推敲支援システムの多くは，字面処理や形態素解析レベルの処理により文法中心の誤り検出，あるいは誤り訂正を行なうことがその支援目標となっている．一方，近年の日本語ワープロの機能強化において，動詞の格フレームと名詞意味素性（プリミティブ）としてのシソーラス(thesaurus)を利用した変換候補の絞り込み手法が実用化されはじめた．そこで本稿では，この手法を校正支援に応用することにより，我々が以前から提案している日本語文法チェッカでは検出できなかった校正・推敲対象の選択手法として，特に，従来の文法チェッカの高速性を損なわず，意味解析の初段に相当する処理－シソーラスチェッキング－の実現手法について述べる．

2．校正・推敲支援

表現された日本語文書（べたテキスト）中から文法情報や表現に関する統計情報の取得を行なうと同時に，最初に規定された構文規則から外れる文の検出を行ない，訂正を促すための適切なメッセージをユーザに通知すると共に，そのための編集操作環境を提供するものを校正・推敲支援ツールと呼ぶ．我々は，字面情報を利用して形式文節を高速に抽出し，文節単位の解析を目的としたPCレベルで稼動するソフトウェアの開発を行なってきた[1]～[3]．このツールは，日本語文法チェッカHSP(High-Speed Proofreading tools)と名付けられており，現在，MS-DOS, MS-Windows上にインプリメントされている．

HSPでは，全角文字列を処理対象とし，形式３つ組み文節（JFK構造）を抽出する．JFK構造では，入力となる表層の文字種別情報から自立部(J部)，付属部(F部)，句読点部(K部)を認識することで１つの形式文節を決定する．

形式文節		
非ひらがな列	ひらがな列	句読点記号
自立部(J)	付属部(F)	句読点部(K)

図1.JFK構造

決定された文節毎に，J部，F部の文字列接続状態を，自立語辞書，および接続行列を用いて検定を行なう[3]．再帰的最長一致検索によりJ部を処理し，自立語辞書に存在しない部分文字列を発見した場合，J部に誤りが混入したものとする．また，助詞，助動詞，補助動詞，補助体言などの付属語要素については，付属語構成要素としての部分文字列間の隣接ビット行列により表現された接続行列を用いてF部の誤り検出を行なう．

ここまでの手法では，形式文節毎の形態素レベルでの統語的な誤り検出ができるに過ぎない．すなわち，より強力な校正チェックを実現するためには，さらに工夫が必要となる．

そこで，格文法解析で一般に使用される格フレームと名詞意味素性の利用について検討した．これらの格文法要素の利用により，シソラースチェッキングを行なうことが可能となる．

3．動詞格フレームと名詞シソーラス

3.1 動詞格フレーム

日本語の形式文節のうち，用言を修飾するものを格(case)と呼ぶ．また，格によって修飾される用言を述語(predicate)と呼ぶ．ここで，述語のうち動詞のみをターゲットとして考える．

動詞を述語として考えた場合，この述語に関係する格は，必須格(obligatory case)と任意格(optional case)とに分類することができる．この必須格集合を格フレーム(case frame)と呼ぶ．

一般に格は自立名詞類と格語尾(case ending)との組で表現される．また，自立名詞類を名詞概念として捉えることで，名詞概念に相当する意味の階層を名詞シソーラスで表現できる．

以上から文(sentence)の簡単な定義をBNFによって示す．

```
1:  <Sentence> ::= <Cases><Predicate>
2:  <Cases> ::= <Case>*
3:  <Case> ::= <Noun-concept><Case ending>
4:  <Noun-concept> ::= {名詞類}
5:  <Predicate> ::= {動詞類, 形容詞, 形容動詞}
6:  <Case ending> ::= {助詞類}
7:  <Case ending> ⇒ <Predicate>
8:  <Noun-concept> ⇒ F(<Predicate>, <Case ending>)
```

図2.格と述語による文の定義

ここで，記号⇒は，ターミナルシンボル（終端記号：terminal symbol）への依存関係を示す．また，関数Fは，複数のターミナルシンボルのコンボリューションを示す．また，規則７は，格フレームの定義と見なすことができる．さらに規則８は，シソーラス定義と見なすことができる．

3.2 名詞シソーラス

ターゲットとなる述語と文中の格との接続関係は，表現された文の意味解析的な検定結果を左右する．このことにより，意味解析の初段に相当する校正チェックを考えることができる．

一般に，かな漢字変換時における誤り混入は，文節区切りの間違いによるものや，同音異義語の選択誤りが多い．

これらの誤り検出のためには，自立語のうち名詞をターゲットとするシソーラス辞書を用いたアプローチがAI変換技術として知られる．実用レベルでは，一般に特定の自立語と述語との関係を用例辞書という形で提供する．変換精度の向上を期待する場合，用例数が10～50万程度と言われている．

Development of Proofreading Tools for Japanese Document
-Using Verb Case Frames and Noun Thesauruses-
Kazuhiro NOTOMI, Shinji MASUDA, Tatsuya KATO,
Satoshi OHNO and Akihiko UCHIYAMA
School of Science and Engineering, WASEDA University

名詞シソーラスは，木構造(tree)として表現される．自立名詞類のインスタンス(instance)がリーフ(leaf)に，また，名詞概念あるいはクラスがノード(node)にマッピングされる．ひとつの名詞インスタンスの意味階層は，ルートからリーフへ至るシソーラスパス(thesaurus-path)として表現される．

4．アルゴリズム

4.1 格フレームチェッキング

格フレームチェッキングでは，文から抽出された形式文節（JFK構造）をターゲットにチェックを行なう．述語要素の決定後，前置された文節群から格文節を抽出し，これが述語要素の格フレームに含まれるか否かを判定する．格フレームは，格語尾（助詞類）により判定できる．判定結果が偽であれば，メッセージを送り，修正編集を行なうかをユーザに通知する．C++による擬似コードを以下に示す．

```
void    Caseframe::Checking(JFK& Snt)
{
    int  i = Snt.CaseNum;
    if(! i)    return;
    while(--i){
        if(IsPred(Snt.Case[i]) == TRUE){
            int  j=i;
            if(! j)    return;
            while(--j)
                if(! IsCaseOf(Snt.Case[j], Snt.Case[i]))
                    Message("Encountered an error !");
                else NounThesaurus::Checking(Snt, j, i);
        }
    }
}
```

図3.格フレームチェックメソッド

4.2 シソーラスチェッキング

格フレームによるチェック結果が真であっても，述語が要求する名詞概念を持たない自立名詞類が使われていた場合，これをエラーとしなければならない．例えば，述語「食べる」の対象格(obj)が「コンピュータ」であれば，「コンピュータを食べる」という表現になるので，明らかな慣用表現以外では，エラーとして扱い，校正対象に加える必要がある．

格フレームは，格語尾の他に名詞概念を持つのでこれを取得する．例えば，述語「食べる」の格フレーム情報のひとつは，

述語::格標識(名詞概念，格語尾)＝
食べる::Agent(**life**, が)

のように表現される．ここで life が名詞概念である．そして名詞シソーラス辞書からフルパスでシソーラスパスを取得する．例えば，インスタンス「太郎」のシソーラスパスは，

「\object\life\animal\human\male\boy\太郎」

のように表現される．

シソーラスチェックでは，格フレームから取得した名詞概念情報と，シソーラス辞書から取得したシソーラスパスとのマッチングを行ないエラーの判定を行なう．C++による擬似コードを以下に示す．

```
void  NounThesaurus::Checking(JFK& Snt, int j, int i)
{
    char*   Pred = (char*)Snt.Case[i];
    char*   Noun = (char*)Snt.Case[j];
    char    NConcept[STRSIZ], Path[STRSIZ];
    GetNounConceptFromCaseFrame(NConcept, Pred);
    GetThesaurusPathFromDic(Path, Noun);
    if(! strstr(Path, NConcept)){
        Message("Encountered an error !");
    }
}
```

図4.シソーラスチェックメソッド

4.3 文法チェッキングとのリンク

もともとHSPは，文法チェッカであり，HSPの校正・推敲対象検出能力の向上を目指す１つのアプローチが以上述べてきた格フレームチェッキングとシソーラスチェッキングである．従って，従来のプロセスにこれらを組み込む必要がある．文法チェッキングは，JFK構造の生成からはじまり，J部,F部の検定を実行する．この処理が終了した時点で格フレームチェック，シソーラスチェックのメソッドを起動する．C++による擬似コードを以下に示す．

```
void  Grammar::Checking(char* Sentence)
{
    JFK     Stn(Sentence);
    BOOL  Err[3];
    Stn.MakeJFK();
    int  i = 0;
    while(i < Stn.CaseNum){
        if((Err[0] = TryBoth
            (Stn.Case[i].JPart, Stn.Case[i].FPart))
                != TRUE){
            Err[1] = TryWithJiritsuDic
                        (Stn.Case[i].JPart);
            Err[2] = TryWithConnectionMatrix
                        (Stn.Case[i].FPart);
        }
        if(Err[0] || Err[1] || Err[2])
            Message("Encountered an error !");
        ++i;
    }
    //   Link to Caseframe and Thesaurus Methods
    Caseframe::Checking(&Stn);
    delete  Stn;
}
```

図5.メインメソッド

5．まとめ

動詞格フレームと名詞シソーラスを利用した文書校正・推敲支援手法について述べた．本手法に見られるように，比較的単純な方法で文法レベルでは検出できない表記上のエラーに対処することができるが，①述語要素に助動詞類が含まれた際の格の転移によるノイズへの対処，②格構造の交差が生じた際の格文節抽出エラーへの対処，などの問題点が残っている．これらについては今後検討していく予定である．

文 献

[1]納富，内山，他：知的ワードプロセッサにおける文脈情報の利用，第43回情処全大，(1991.10).

[2]納富，内山，他：日本語文書校正支援ツールの開発ーマニュアル作成支援についてー，第45情処全大，(1992.10).

[3]納富，内山，他：日本語文書校正支援ツールの開発ー解析手法の検討と評価ー，第46情処全大，(1993.03).