ルールベースの優先解釈と曖昧性解消

3M－2

福持 陽士　鈴木 等　九津見 毅
(シャープ株式会社　情報商品開発研究所)

1. はじめに

本稿では、英日機械翻訳システム DUET Qtの新しい翻訳エンジンに採用したパーザ、優先解釈部、遅延意味解析についてその概要を説明し、人間が持つ文に対する読みの選好を機械翻訳システムにインプリメントした方法について述べる。

2.パーザ

本システムで採用しているパーザは、冨田パーザに機能拡張を加えた一般化ＬＲパーザである。機能拡張は、次の２点である。ＬＲ法では、文法をプリコンパイルしてＬＲ表を求めることにより、実行前にパーザの可能な動作を決めておくように設計されている。英語の解析においてよく問題となるギャップ処理にスラッシュカテゴリを使用するためには、ＬＲ表に反映される文法が必要である。このため、本システムでは、下記のような拡張文脈自由文法をプリコンパイルする際に、スラッシュカテゴリ用の５つの文法を自動的に生成しＬＲ表を作成する。

親規則： S → NP VP (HEAD:2 SCORE:50 R1:SUBJ R2:FINITE L: SUBJ-PRED-AGR SLASH:1,2)

子規則： S/NP→VP　S/NP→NP VP/NP　S/ADJP→NP VP/ADJP　S/ADVP→NP VP/ADVP　S/P→NP VP/PP

また、一般化ＬＲ法では、シフト操作時に、現在どの文法規則を適用中であるかは認識されない。文脈自由文法の右辺及び左辺に補強されたチェックは、例えば右辺１項目を処理中に枝刈りできるものもリジュース操作時まで待たなければならないため、無駄な解析木を作成することになる。このため、本システムでは、ＬＲ表にシフト情報をセットする際、そのシフト情報に関連するすべての文法規則番号と文法規則の右辺何項目からのシフト操作であるかという情報をセットするようにした。これにより、解析時の特定シフト操作時のチェックがすべて失敗するような場合、そのシフト操作は破棄され、比較的早い時点で解析木は枝刈りされることになる。

3. 構文優先解釈モジュール

本システムのパーザが出力する複数の木構造の優先度決定には、以下の３つの情報が利用される。

・ルール点数：文法に与えられたスコアから算出される尤度
・語彙点数　：木構造を構成している語彙項目に対して辞書で定義された尤度
　例　pen (= a writing tool): Normal
　　pen (= to write):　　Minor
・構造点数　：構文優先解釈規則により与えられる尤度（ルール点数に掛けられる）

また、本システムの構文優先解釈規則は、下記の記述形式により記述され、文の大域的構造を参照

Rule-Based Preferential Interpretation and Ambiguity Resolution
Yoji Fukumochi, Hitoshi Suzuki, Takeshi Kutsumi
Sharp Corporation, Information Systems Product Development Laboratories
492 Minosho-cho, Yamatokooriyama, Nara 639-11, Japan

して、特定の解釈の優先度を下げたり、あるいは、上げたりする枠組である。

MATCH: T: tree-match-pattern
IF: T.n.property = value あるいは T.n1.property = T.n2.property
COND:T.n.procedure-name
THEN:weight-value

IF部にかける属性は、SYN(統語的属性)、SEM(意味属性)、SUB(細分類)、LEX(見出し)などである。簡単な記述例として、次のようなものがある。

MATCH: T: NP(NP(NP,PP),CONJ,NP(NP,PP)
IF: T.4.LEX = T.8.LEX
THEN: 1.2

"A of B and C of D" という単語列に対しては、①((A of B) and (C of D)) ②(A (of ((B and C) of D))) ③(A (of (B and C)) (of D)) ④(A (of (B and (C of D))))の４つの解釈が可能であるが、意味的に有意な情報がなければ①の解釈を取る可能性が高い。このような人間が持つ文の大域的な構造に対する選好を記述したものが本システムにおける構文優先解釈規則である。この規則は、パーザがリジュース操作を実行する度に照合が取られ、マッチした場合、木構造の優先度が変更されるようになっている。

４．遅延意味解析

本来、パーザが出力する構文木の意味的整合性の検査は、すべての可能な解釈に対して行い、検査に合格した候補の中から最良のものを選択するというのが理想的な処理形態であろうと考えられるが、実用的なシステムの構築においては、すべての packed node を展開することは、処理手数が膨大なものとなり、その処理に払うコストは、非常に expensive なものになってしまう。このため、本システムでは、構文解析中には、一切、意味解析を行わず、純粋に統語的な情報のみ利用して、候補の優先度を仮決定する。構文解析終了と同時に、最良候補として選択された候補に対して、意味解析を実行し、その検査結果に何らかのマイナス要因が含まれた時初めて、packed node の展開と第２次候補に対する意味解析を行うよう設計されている。意味解析におけるマイナス要因は、例えば、①意味解析不合格、②意味解析は、デフォルトの格パターン(意味制限を解除した格パターン)を採用して合格した、③意味解析は、マイナーな格パターンを採用して合格したなどである。本システムでは、これらの意味解析におけるマイナス要因が検出されると、検出された単語位置を含む最小packed nodeが展開され、次に最良の候補が出力される。

５．おわりに

優先解釈規則という選好の情報を解析文法に記述された制約とは切り離したルールベースの形で蓄積していこうというのは、我々にとって初めての試みで、今後の我々の研究において

・競合する規則を追加しようとした際、警告を出すなどの規則の整合性を保守するツール
・優先解釈規則を大規模なコーパスから、半自動的に獲得するためのツール

の開発などが重要と考えられる。これらを加味しながら、今後更に実験を繰り返すことにより、翻訳の精度を向上させていく予定である。