定型利用翻訳における構造解析精度の評価

富士 秀、潮田明

（株）富士通研究所

fuji.masaru@jp.fujitsu.com

1. 概要

機械翻訳が長文の翻訳に弱いという問題を解決するため、我々はこれまで、原文の定型性を利用して原文を部品に分割してから機械翻訳を適用する「定型利用翻訳」の研究を行ってきた[1][2][3]。本研究では、対象文種として特許抄録文を取り上げ、特許抄録文から定型パターンを抽出してこれを定型利用翻訳システムに組み込むことによって、対象文種に対して高精度に構造解析が行えるようにした。この対象文種にチューニングされた定型利用翻訳の構造解析精度の評価を行い、さらに、従来型の構文解析エンジンの解析精度と比較することによって、定型利用翻訳の導入による解析精度向上の効果を定量的に測定したところ精度向上が確認された。

2. 背景

長文における機械翻訳では、原文構造を正確に把握し、この原文構造に基づいた訳文を生成する必要がある。しかしながら従来型の機械翻訳システムでは、局所的な処理が中心となっているため、長文の全体構造を高精度で把握することができない、という問題があった。

多くのルールベース機械翻訳では、構文解析においてボトムアップ的な処理がベースとなっているため長文の全体構造を把握して処理することはできない。また統計ベース翻訳も、対訳コーパスにおける隣接する単語の統計情報がベースとなっているため、局所的な処理の積み重ねには変わりなく、入力文全体の構造を正確に訳文に反映させることは困難である。

このような従来型機械翻訳の弱点を克服するために、我々は、長文の定型パターンを利用して、文全体の構造を高精度に解析する翻訳システムである「定型利用翻訳」を開発してきた[1][2][3]。

3. 今回の実験

今回の実験では、産業分野において膨大な量の翻訳作業が行われている特許文が多くの定型パターンを含んでいることに着目し、特許抄録文から定型パターンを抽出して定型利用翻訳システムに搭載することによって、定型利用翻訳を特許抄録文にチューニングした。

定型利用翻訳は、定型パターンを利用して日本語原文の構造を解析する日本語構造解析（SAS）エンジン[1]と、日本語構造解析結果から英語訳文を生成する生成エンジン[3]から構成されるが、今回の実験は日本語構造解析（SAS）エンジンのみを実験・評価対象とした。

比較対象として、従来のルールベース機械翻訳で用いられる構文解析モジュールと同等な機能を持つ、フリーソフトウェアとして公開された構文解析エンジン CaboCha を用いた。

以上の実験構成を図 1.に示す。

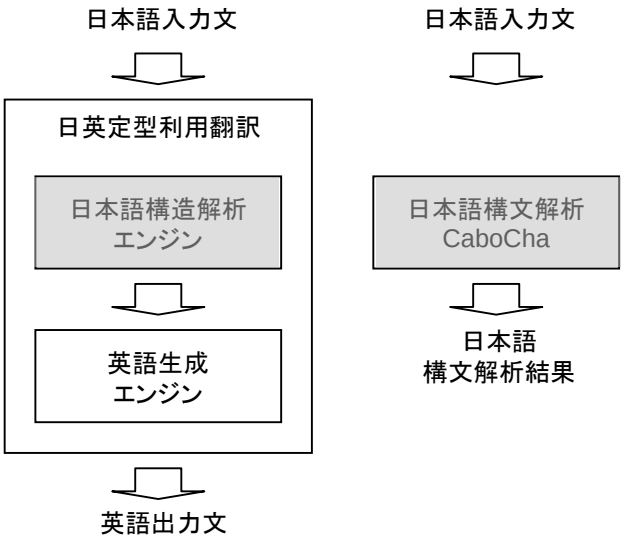

図 1. 実験構成図

4. 日本語構造解析（SAS）エンジン

4.1. 入出力

日本語構造解析（SAS）エンジンは、対象分野の定型パターンをあらかじめ内部のデータベースに蓄えておき、日本語文が入力されると、適用できる定型パターンを検索し、検索された定型パターンに沿って入力文を分割して出力する。今回は、特許抄録文から抽出した定型パターンをデータベースに蓄えた状態で実験を行った。

図 2.は、日本語構造解析（SAS）エンジンへの入力である、日本語特許抄録文の例文である。図 3.は、定型パターンのデータベースであり、日本語特許抄録文から抽出した定型パターンを格納している。図 2.の入力文と図 3.の定型パターンデータベースを照合したところ、定型パターン P53 がマッチすることがわかった、とする。このマッチ結果から、入力文は図 4.のように分割されて出力される。

Copyright(C) 2012 The Association for Natural Language Processing.
All Rights Reserved

この遊技機は、遊技開始を要求する入力を受け付けるスイッチと、遊技開始を要求する入力に応じて、内部当選役の決定を行うメインＣＰＵと、待機時間の開始からの経過時間を計時する待機時間経過タイマＩＣと、メインＣＰＵが決定した内部当選役に応じて、内部当選役を報知させるサブＣＰＵとを有する。

図 2. 入力日本語文

P52：主題　主動詞
P53：主題　目的*　主動詞
P54：主題　目的*　主動詞　主題
P55：修飾節　主題　主動詞
P56：修飾節　主題　目的*　主動詞

図 3. 定型パターンデータベース

ラベル	構造部品
主題	この遊技機は、
名詞句	遊技開始を要求する入力を受け付けるスイッチと、
名詞句	遊技開始を要求する入力に応じて、内部当選役の決定を行うメインＣＰＵと、
名詞句	待機時間の開始からの経過時間を計時する待機時間経過タイマＩＣと、
名詞句	メインＣＰＵが決定した内部当選役に応じて、内部当選役を報知させるサブＣＰＵとを
主動詞	有する。

図 4. 日本語構造解析（SAS）エンジンの出力

4.2. 動作

入力文と定型パターンのマッチングは、文節レベルで行う。これは、日本語解析では、長文であっても、文節解析までの解析精度は高くほぼ確実な結果が得られるからである。

さらに、定型パターンと文節列の照合では、探索効率を上げるために「特徴文節」を用いている。特徴文節とは、対象文種において頻出する特徴表現[1]を含む文節を定義したものである。定型パターンは、これら特徴文節とのみマッチするようになっており、これによって探索空間が圧縮できる。

また、入力文に対して複数の定型パターンがマッチした場合には、各定型パターン候補に対して、評価値を付与するための枠組みを用意した。例えば、文全体としてバランスの良い定型パターン候補には高い加点が与えられるようなヒューリスティックスを導入している。

4.3. 定義ファイル

定型利用翻訳では、対象文種毎に以下で述べる２種類の定義ファイルを作成する。対象文種へのチューニングは、対象文種に対する定義ファイルの内容を調整することによって行う。

定型パターンファイル

定型パターンは、文全体のもっとも外側の構造である第１階層の構造を記述したパターンである。具体的には、文全体の主題、この主題を直接の主語とする主動詞、この主題に直接係る修飾句、主動詞の目的語となる名詞句、を構成要素とする。

定型パターンは、対象文種毎に作成されるものである。

特徴文節定義ファイル

特徴文節は、定型パターンと同様に、対象文種に応じて作成されるものである。

特許抄録文では、例えば、定義ファイルに「～において、」というエントリーが記述された場合、この文字列を右端とする文節は、特徴文節として定義されたことになる。特徴文節として定義された文節は、定型パターンとのマッチング候補となることを表す。

4.4. 各処理モジュール

文節処理

日本語入力文に対する文節処理では、日本語形態素解析を用いる。入力文に対する形態素解析によって得られた各形態素には、自立語や付属語の属性が付与されるが、この自立語・付属語の情報を用いて、形態素列から文節列への変換を行う。

特徴文節の特定モジュール

入力文中の各文節列と、特徴文節定義ファイルとの照合を行い、特徴文節にマークを付与する。特徴文節として特定された文節は、定型パターンとの照合において、マッチングの候補となる。

定型パターンのマッチング

特徴文節がマークされた文節列と、定型パターン定義ファイルとの照合を行い、マッチするすべての定型パターンを見つける。マッチング結果としては、一つの定型パターンのみが見つかる場合もあるが、複数の定型パターンが見つかることもある。

各定型パターン候補への加点付与

上記マッチングによって得られた複数の定型パターン候補をランキングするために、各定型パターン候補に加点を付与する。より自然なマッチング候補に高い加点が付与されるような枠組みを用意する。例えば、並列構造を持った定型パターンにおいて、各並列要素が同様の右端表記を持っている場合には、その候補に加点を行う。

結果出力

前項によって付与された加点の順に定型パターン候補のソートを行い、その結果を出力結果として表示する。

Copyright(C) 2012 The Association for Natural Language Processing.
All Rights Reserved

5. 実験

5.1. 実験の準備

実験テキストの用意

実験には、2,000 案件の日本語特許抄録文を使用した。各特許抄録案件は「発明の名称」、「課題」、「解決手段」の３文から構成されるが、今回は長文に対する評価を目的としているため、特に長文を多く含む「解決手段」文を用いた。これら 2,000 文の特許抄録「解決手段」文を、①定型パターン作成のための学習セット、②チューニングセット、③評価用セットの３つ（比率は 7:2:1）に分割した。

定型パターンの作成

定型パターンの作成段階では、学習セットの各解決手段文を見ながら定型パターンを人手で抽出し、日本語構造解析（SAS）エンジンの定型パターンデータベースに格納した。およそ 200 案件の学習セット文から、大分類で約 60 種類の定型パターンを抽出した。またこの学習セットを用いて特徴文節の抽出も行った。

日本語構造解析エンジンのチューニング

日本語構造解析（SAS）エンジンのチューニング段階では、チューニングセットの日本語文を入力として、上記定型パターンおよび特徴文節を搭載した日本語構造解析（SAS）エンジンによる解析を実行した。日本語構造解析（SAS）エンジンの出力が正しくなるように定型パターンおよび特徴文節の調整を行いながら前記エンジンによる解析を繰り返すことによってチューニングを行った。

5.2. 評価方法の策定

本実験の評価では、長文における文全体の構造が正しく把握されているかを評価することを主目的としているため、文構造の第 1 階層に絞って評価を行った。また、定型利用翻訳における日本語構造解析の精度と、従来型構文解析（例として CaboCha を使用）の精度が同レベルで比較できるような評価手法を選択した。

図 5.は、入力文の例である。図 6.は、入力文に対する日本語構造解析（SAS）エンジンによる解析結果である。また、図 7.は、同じ入力文に対する、CaboCha による構文解析の出力例である。

この遊技機は、遊技開始を要求する入力を受け付けるスイッチと、遊技開始を要求する入力に応じて、内部当選役の決定を行うメインＣＰＵと、待機時間の開始からの経過時間を計時する待機時間経過タイマＩＣと、メインＣＰＵが決定した内部当選役に応じて、内部当選役を報知させるサブＣＰＵとを有する。

図 5. 入力文

図 6.の日本語構造解析（SAS）エンジンの出力では、定型パターンが、文の第 1 階層に対して作成されているため、出力も第 1 階層の分割に関するものになっている。

ラベル	構造部品
主題	この遊技機は、
名詞句	遊技開始を要求する入力を受け付けるスイッチと、
名詞句	遊技開始を要求する入力に応じて、内部当選役の決定を行うメインＣＰＵと、
名詞句	待機時間の開始からの経過時間を計時する待機時間経過タイマＩＣと、
名詞句	メインＣＰＵが決定した内部当選役に応じて、内部当選役を報知させるサブＣＰＵとを
主動詞	有する。

図 6. 日本語構造解析（SAS）エンジンの出力

図 7.の構文解析出力では、木構造の各ノードが「D」によって表されており、並列ノードは「P」であらわされている。ここで、右端の D ノードおよび右端より２番目の P ノードまでが、文の第 1 階層を表している。

本評価では、この第 1 階層に相当する部分のみを対象とした評価としたため、第 2 階層以下は評価対象外とした。このようにすることによって、図 6.の日本語構造解析結果と図 7.の日本語構文解析結果を同列に比較することが可能となった。

```
      この-D
遊技機は、----------------------------------------------------D
 遊技開始を-D                                                 |
    要求する-D                                                |
       入力を-D                                               |
     受け付ける-D                                             |
       スイッチと、-------------------------------------P     |
        遊技開始を-D|                                      | |
            要求する-D                                     | |
                入力に-D                                   | |
              応じて、-----D                               | |
            内部当選役の-D|                               | |
                    決定を-D                               | |
                      行う-D                               | |
           メインＣＰＵと、----------------------P          |
                    待機時間の-D                           | |
                     開始からの-D                          | |
                       経過時間を-D                        | |
                         計時する-D                        | |
        待機時間経過タイマＩＣと、----------------P        |
                       メインＣＰＵが-D                    | |
                            決定した-D                     | |
                          内部当選役に-D                   | |
                                応じて、---D               | |
                              内部当選役を-D               | |
                                  報知させる-D             |
                                サブＣＰＵとを-D
                                      有する。
```

図 7. CaboCha 形式の構文解析結果例

Copyright(C) 2012 The Association for Natural Language Processing.
All Rights Reserved

5.3. システムの実行

日本語構造解析（SAS）エンジン

評価用セットから 30 文を入力として用いて、対象文種にチューニングされた日本語構造解析（SAS）エンジンを実行した。定型利用翻訳によって出力された構造解析結果に対して人手による正誤判断を行って評価した。

日本語構文解析 CaboCha

前項と同じ評価用セットを入力として用いて、日本語構文解析エンジン CaboCha を実行した。構文解析結果は木構造となっているが、そのうち、第 1 階層に相当する部分に対して人手による正誤判断を行って評価した。

6. 結果

図 8.に実験の評価結果を示す。

縦軸は正解率であり、評価用セットの 30 文のうち、正解となった文の割合を表している。

入力文はもともと長文で構成されているが、その中でも特に、より長い文（文節数 41 から 80）と、それよりは短い文（文節数 11 から 40）の二つに分け、それぞれについて解析精度のプロットを行った。

日本語構造解析（SAS）エンジンは、複数の候補を出すことができるが、1 位候補の正解のみをカウントしたのが「1 候補」と表示されているグラフであり、第 1 位および第 2 位のいずれかに正解が含まれる場合にカウントしたのが「2 候補」である。

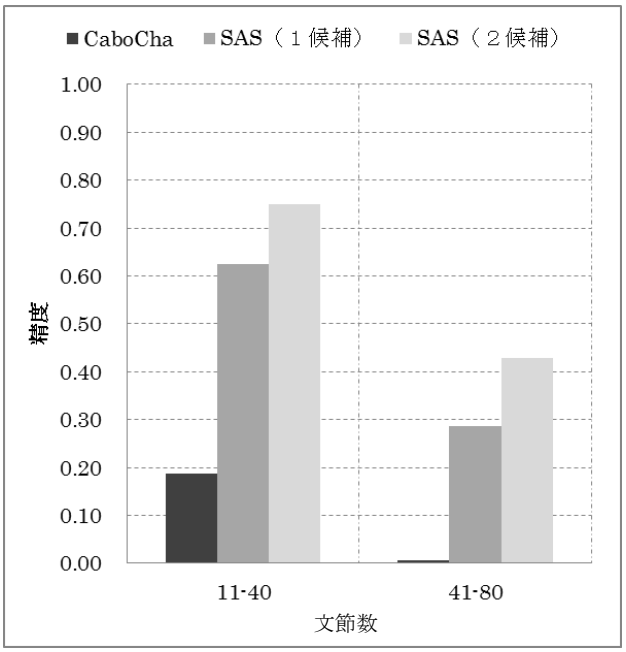

図 8. 文節数による構造解析精度の比較

7. 考察

全体として、対象文種にチューニングされた日本語構造解析（SAS）エンジンは、日本語構文解析 CaboCha より高い解析精度を示している。精度向上の傾向は、比較的短い入力文のグループでも、より長い入力文でも、両方で認められる。

なお、文節数 41 以上のより長い入力文のグループでは、従来の日本語構文解析では正解が得られていないが、日本語構造解析（SAS）エンジンのほうでは一定の正解率を出すことに成功している。

8. まとめと今後

従来型機械翻訳が長文において構造解析精度が低いという問題を解決するために、定型利用翻訳を開発した。今回対象文種として取り上げた特許抄録文の定型パターンを日本語構造解析（SAS）エンジンに搭載することによって、対象文種へのチューニングを行った。その結果、対象文種へのチューニングによって、従来型構文解析と比較して、第 1 階層の解析精度が大幅に向上することがわかった。

長文における第 1 階層の解析精度向上は、長文に対する機械翻訳精度向上に向けた大きなステップと言える。今後は、第 1 階層の解析精度向上が、全体の翻訳精度向上にどのようにつながっていくのかを評価していきたい。

また今回は、日本語特許抄録の「解決手段」文を例にとってチューニングを実施したが、今後は、さまざまな分野でのチューニングを行って、精度向上の度合いを比較していきたい。

さらに、現在は人手に頼っている定型パターンの汎化プロセスについて、少しでも自動化に近づけられないか検討していきたい。

参考文献

[1]富士秀, 長瀬友樹, 潮田明, 増山顕成. 定型性の高い文章に対する日本語構造解析.言語処理学会第 14 回年次大会予稿集,2008.

[2]富士秀, 長瀬友樹, 潮田明, 増山顕成. 原文の定型性を活用した機械翻訳精度向上手法.言語処理学会第 15 回年次大会予稿集,2009.

[3]富士秀, 長瀬友樹, 潮田明, 増山顕成. 部品化された原文からの機械翻訳文生成.言語処理学会第 16 回年次大会予稿集,2010.

Copyright(C) 2012 The Association for Natural Language Processing.
All Rights Reserved