産業日本語の確立のための理論と方法

文書記述のための制限日本語

立命館大学情報理工学部情報システム学科教授 **池田　秀人**

PROFILE

30年以上データベース技術の研究を進めてきたが、2004 年ころから自然言語処理の研究を始め、機械翻訳、言語教育 e ラーニングシステム、特許文書作成支援システムなどの研究を行っている。

hikeda@is.ritsumei.ac.jp　　TEL 077-561-2691

1 はじめに

現在の機械翻訳システムにとって、特許文書は厄介なもののひとつとされている。その理由は、「1 文がながいこと」や「同意の多様な表現がある」ことが挙げられる。しかし、「日本語特有の文末表現の種類が少ない」、「専門用語の訳語はほぼ一意に決まる」、「記述内容に制約が強く文の種類が少ない」、「基本的に曖昧性を排除しようとしている」、「普通の日本語では省略されやすい単数か複数の区別、1 以上か 0 を含むかの区別、特定か不特定の区別など英語に翻訳する場合、翻訳者が判断しなければいけない情報を文中に明示している」などの点で、翻訳しやすいという特徴もある。特許文書のこれらの特徴を考え、特許文書記述用の制約言語を作成することは、次の点で有意義である。

- 特許文書の品質を向上させる。（曖昧性の除去や分かりやすさの向上）
- 機械翻訳の精度を飛躍的に向上させることで、特許取得のための時間的・経済的コストを削減する
- 特許検索の精度を向上されることで、特許管理を効率化する

この論文は、このような目的のため、特許文書記述用の制約言語を設計する方法を提案する。

2 特許文書記述用制約言語の背景理論

一般にすべての正しい文の集合（「集合Ａ」とする）は、次のような文のタイプに分けられる。

（1）タイプＦの文：機械翻訳の対象にしたくない文（人種・性差別文、スラング、極めて制約されたグループ間でのみ通用する文など）

（2）タイプＤの文：意味的にも構文構造的にも同じ文が翻訳しようとしている言語にある文

（3）タイプＣの文：Ｄ内の文と同じ意味を持つ自然な文（構文構造は異なってもいい）

（4）タイプＢの文：他の言語に翻訳するには、情報が不足していたり、曖昧性を持っていたりする文

これを図示すると図１のようになる。

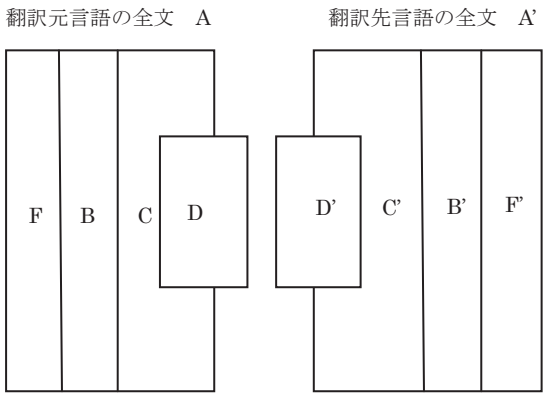

図１　さまざまな文集合

ここで、それぞれの集合の典型的な文を示そう。

- タイプＢの文：「その後、フォトレジスト１９を除去する」。この文の基本形は、「（誰）が（物１）を（物

S0=_ は _ を _ 含んでいる ([N1],[N2],[N3]); N1= この _([N: システム]); N2=__([P4],[N: 要素]); N3=_ 以上 ([N: 3つ]); P4=_ を満足する ([N: 性質 B]);	S0=_-@include:p1-_-_([N1],[N2]); N1=this-_([N:system]); N2=_-_([N:component],[P4]); N3=more-than-_([N:3]); P4=having-_-([N:property B])

図2　構文構造の異なる翻訳文の例

2）から除去する」である。問題の文は、基本形の「誰」と「物1」が省略されている。副詞「その後」から、前文から「物1」が推測されるから書かなくてもいいと考えているのだろう。また、「誰」は特に明示しなくても「オペレータ」であることが分かるから明示していないのだろう。しかし、この文を独立した文として見るとき、翻訳するのに必要な情報が無い文すなわちタイプBの文ということになる。

- タイプC：の文「このシステムは性質Pを満足する要素を3つ以上含んでいる」。この文は、”This system includes more than 3 components having property P.”の日本語訳として使われている。しかし、この日英の文は構文構造が同じではない。これらの文を構文構造に分解してみると、図2のようになる。この構造を持った（タイプDの）文は、「このシステムは性質Pを満足する3つ以上の要素を含んでいる」とすれば、可能であるが、日本語文としては前者の方が自然で、実際特許文書中でも頻出する。従来の機械翻訳は、これらの文のタイプを無視して、すべての文を翻訳しようとしていた。これが翻訳品質の悪い1つの原因である。

3 制限言語としての産業日本語

制限日本語を設計する場合必要なことは、「曖昧性を含まない理解しやすい文の基準を見つけること」、「冗長性や言い換えを含まない最小規模の言語体系であること」、「機械翻訳で正しい翻訳ができること」等が求められる。それを満足する体系を次のように構築することを提案する。

- （語彙・文型制限）語彙だけでなく文型も制限すること、
- （内容非制限）必要な内容の文を書くことができること
- （判定可能性）与えられた文が、この制限言語の範囲内に入っているかどうかが容易に判定できること、
- （非文排除）文法的な誤りが無い文のみでできていること、
- （表現最小性）同じ内容を持つ複数の文を生成しないようにすること、
- （翻訳品質保証）与えられた文が他の言語に機械翻訳できることが保証されていること

この制約を満たす言語を設計するには、文型制約が重要な役割を果たすが、文型をすべて列挙することは数が多すぎて現実的ではない。しかし、用語の制限だけでは、「判定可能性」や「非文排除」の条件を満たすことができない。そこで重要なのは、「句型制限」と「語句間共起制限」である。

4 句関数を使った産業日本語の定義

4.1 句関数の定義

句関数とは、図2のような文字列関数で、関数名に語句以外に引数語句を埋め込む位置や変換方法が明示されているものである。関数の実行結果は、N（名詞句）、P（述語）、S（文）、C（主格補語）の4つのタイプがある。関数名にはその関数の意味を表す中心語が含まれることが多い。引数には、その関数の中心語と引数の関係を示す関係子（object, place, time等）がついている。また、

関数全体の意味を示す属性 (アスペクト、モダリティ) がついている。

例えば、上の図２の

S=_ は _ を _ 含んでいる ([N1],[N2],[N3]);

という関数の関数名は、「_ は _ を _ 含んでいる」であり、引数の位置は下線（_）で示されている。また引数Ｎ１～３は、それぞれ agent, object, quantity という関係子を持ち、関数全体は、モダリティ state を持っているので、関数辞書の中には、

S=_ は _ を _ 含んでいる

([N: agant],[N2:object],[N:quantity])

{ state };

として格納されている。この関数に具体的な引数の値を入れて実行されると、文ができるから、この関数のタイプは S(文) である。ここで句と呼んでいるのは、文や語も含んでいる。この論文で提案する制限日本語は、使用できる句関数と埋込語句を制限し、更に語句間共起制限を加えたものである。

４.２　表現最小化のための標準語彙・関数

句関数制限で、表現最小性と満たすために、同義語の中からもっとも標準的なものを１つ決め、他の語は常に標準用語に一旦置き換えてから翻訳プロセスに入ることとする。この手法は、述語や文の関数にも使え、内容表現力を低下させずに語彙や関数の数を少なくすることができる。

たとえば、表1のような同じモダリティ「appearance」を表す文末表現を見ると、表の第３列のような多様な表現がある。この文末表現を置き換えて自然な文になるのは、「ようである」だけで、他のものは、不自然な文を作る。従ってモダリティ appearance を表す関数としては、

S=_ ようである ([S]){appearance}

だけで十分だということになる。

このような現象は、アスペクトや形容詞関数、副詞関数にもあり、このような方法で表現内容に制約を加えないで関数の数を減らすことができる。

４.３　語句間共起制限

引数となる語と句関数を、すべての内容の文が合成できるように制限（句型制限）しただけでは、非文排除条件や翻訳品質保証条件を満たすことはできない。そこで重要なのは、「語句間共起制限」である。これは、特定の関数にどんな語句や関数を引数としてとることができるかを示したもので、具体的な語句や関数間の関係として表現される。

表１

モダリティ	文	文末表現
appearance	この両者のイメージは結び付かないようである。	よう である
appearance	ずっとヒヒと暮らしているから兄弟みたいなものだ。	みたいな
appearance	どうやら取り出すのを断念した様子であった。	様子だ
appearance	どうやら切符に書き込むボールペンがないらしい。	らしい
appearance	案内しながら語るヨーゼフさんは満足そうだった。	そう
appearance	火災報知器は誤作動だったと見られる。	たと見られる
appearance	企業サイドも大半は模様ながめというところだ。	というところだ
appearance	赤や黄など原色の織物がまぶしいほどだ。	ほど
appearance	池田は３回連続無投票の公算が大きい。	公算が大きい
appearance	透明でありながら、透明じゃないみたいな。	みたいな

4.4　言語内変換による翻訳品質の保証

第３節で示した文のタイプで、タイプDの文は構文ごとに翻訳できる文であるが、これが自然な表現になっていることは保証できない。自然な文から自然な文への翻訳を保証するためには、タイプCの文からなるものにしなければならない。タイプCの文は、意味が同じタイプDの文に一意に射影できるから、Dを介して目的言語のタイプDの文に翻訳できる。しかし、これが目的言語内のタイプCすなわち、自然な文になっている保証はない。タイプDの文からタイプCの文を一意に決めるには、実は分野、話者、話者聞者関係、文体などの限定（個性限定）が必要になる。この論文では、これを特許という分野に限定して行おうとしている。

言語内文変換を行うために有効なのは、「言い換え表現データベース」である。この提案では、Alagin が提供するデータベースを使って、「言い換え表現辞書」を作り、同じ意味の表現の中から１つの標準形を決めて制約を実現しようとしている。

例えば、＜AはBの原因となる＞という表現は、<Bの原因である A>、<A はB の原因にもなる >、<Bの原因となる A>、<B の原因になる A>、<B の原因にもなる A>、<A がB の原因となる >、<B の原因とされる A>、<A がB の原因になる >、<A はB を引き起こす >、<B の原因は A です >、<A がB の原因 >、<B などの原因となる A>、<A が原因で起こる B>、<A が原因となる B>、<A がB を引き起こす >、<A はB の原因です >、<B の原因 A>、<B の原因は A> 等の言い換えが可能であるが、これを＜AはBの原因となる＞を標準型として１つ定めることにより、表現最小性を実現している。

5　産業日本語の利用のための支援

制限言語としての産業日本語が受け入れられるためには、次のことが必要不可欠である。

- （自動変換システム）普通の日本語を同義の産業日本語に自動変換するシステム
- （入力支援システム）産業日本語で文を作成することを支援するシステム

自動変換システムは既に作成されている文書を、ここで規定した産業日本語に変換するためのシステムで、一種の言語内翻訳システムである。このシステムの実現のためには、与えられた文を登録してある関数の列で表現するプログラム（関数化プログラム）が必要となる。この関数化プログラムは、一種の構文解析プログラムで、文をトップダウンで解析するピーリングアルゴリズムで実現できる。

入力支援システムは、与えられた関数だけで、文を作成するための支援プログラムで、新しく文を作成する場合、このプログラムを使って文を作成すると、産業日本語で文が作られるとともに、品質が保証された翻訳ができる。このプログラムを実装するために開発した、「自然入力法（普通のワードプロセッサの行っているような文を左から右に入力していき、システムはバックエンドでこれをモニタし、標準表現に変換したり、ポップアップメニューで支援したりする方法）」が有効である。

6　おわりに

ここで紹介した産業日本語確立のための取り組みはまだ始まったばかりで、これから関数辞書の作成や関数の言語間対応、共起辞書を整備していかなければならない。当面、特許分野に個性限定してこれを行っていく計画である。