頑健な文脈処理を取り入れた機械翻訳

那須川 哲哉
日本アイ・ビー・エム株式会社 東京基礎研究所
nasukawa@trl.ibm.co.jp

1 はじめに

高度な自然言語処理を実現するために文脈情報の参照が必要となることは広く認識されており，文脈情報を扱うための様々な手法が提案されてきた [1, 2]．しかし従来の文脈処理手法は，必要となる知識の確保の問題や処理の複雑性などから，実験の域をなかなか越えられずにいる．これは従来の手法が内容理解に踏み込んだ深い処理を基本としているためである．ところが，実用性を重視してそのような深い処理を避け，処理内容を単純なものに限定しても，文単位から文脈単位へと処理の枠組を拡張することにより，様々なメリットを得ることができる．筆者らは，自然言語を処理する際に，複数の文を一括して処理することで入力文全体の解析精度を向上させられることを示してきた [3, 4, 5, 6]．本稿では，機械翻訳において，処理の枠組を文単位から文脈単位へと拡張することで，頑健性や処理効率を犠牲にすることなく，翻訳品質を様々な点で向上させられることを示す．まず次節で単純な文脈モデルを利用した文脈処理の概要を示した上で，続く節において，この処理を実現した英日機械翻訳システム上での実際の翻訳結果を用いて，その有効性を示す．

2 文脈処理の枠組

一般に文脈処理という場合，文脈中で言及されている複数のイベントの相互関係を明確にしたり，文の記述内容を世界モデル中で位置付けるなど，対象世界に関する十分な知識を前提とした深い処理 [1, 2] をさすことが多い．ところがそのような処理は，例えばスクリプトのような文脈処理用の特別な知識を必要とするため，多様な文章の入力を前提とする実用的なシステムへの適用は，(予め十分な知識を用意しておくことが困難なことから) 難しい．そこで以下では，そのような問題を回避するために，文脈処理用の特別な知識を必要としない，極めて表層的で単純な文脈処理の枠組を示す．

まず，文脈を基本的には単なる文の集合体として捉え，文脈中の各文に対する解析処理は，従来の単文処理で行なう構文解析・意味解析程度にとどめる．すなわち文脈モデルは，文中の各文に対し形態素解析及び構文解析を適用することで得られる文中各語の語基や品詞，語と語の依存関係などの情報に加えて，多義性解消などに関する知識が存在する範囲内で知識処理を適用した結果を含めたデータを文脈中の出現順序に従って蓄積したものとする．本手法で構築し参照する文脈モデルの例を図1に示す．このように，文脈モデルは基本的には構文解析レベルで得られる情報を蓄積しただけのデータにすぎず，入力文が構文解析器の文法では完全に解析できない文法的不適格文の場合や，構文解析結果に曖昧性が存在する場合にも，不完全性や曖昧性を残したままでデータを蓄積するため，従来の単文処理の枠組に何ら特殊な処理を加えることなく文脈モデルを構築することができる．文章を処理する際には，まず文章中の各文を解析し，曖昧性を解消せずにそのまま蓄積することでこの文脈モデルを構築し，文脈モデルを参照しつつ，各文の曖昧性解消の処理を行なっていくのが本手法の基本的な枠組である．

文脈モデルは非常に単純な構造しか持たないが，各文の曖昧性を解消する際には，そこに含まれる様々な情報を利用することができる．この文脈モデルを参照することにより，例えば，表層的に同じ語基を持つ語が文脈内では同じ語義を取り同じような語と係り受けを結ぶように各文の曖昧性を解消することで，曖昧性解消の精度を向上させたり [3, 4]，文脈中でうまく構文解析できた文 (適格文) の情報を参照して構文解析が不完全な文 (不適格文) を再解析し解析精度を向上させることができる [6]．また，文脈モデルから得られる語の出現頻度や語と語の依存関係の情報などを参照することで代名詞の照応先決定の精度を高めることもできる [5]．[1] さらには，文脈中の前後の文の構造を参照することで，

- also, only などの副詞の修飾範囲の認識
- 省略の補完
- 箇条書きなどで主語の省略されている動詞句の叙法の判別

なども可能になる．

[1] 技術文書中では 90%を越す精度を得ることができた．

$Discourse = \{Sentence_1, Sentence_2, ..., Sentence_n\}$
$Sentence_i = \{Word_{i1}, Word_{i2}, ..., Word_{ij}\}$

```
John likes apples.        Sentence1:
                           Word1-1[John]  Word1-2[likes]  Word1-3[apples]
 likes                        POS:N          POS:V           POS:N
  ├─(SUBJ)─John             BASE:john      BASE:like       BASE:apple
  └─(OBJ)─apples              :   :          :   :           :   :

Tom also likes apples.    Sentence2:
                           Word2-1[Tom]  Word2-2[also]  Word2-3[likes]  Word2-4[apples]
 likes                        POS:N          POS:ADV         POS:V           POS:N
  ├─(SUBJ)─Tom              BASE:Tom       BASE:also       BASE:like       BASE:apple
  ├─also                      :   :          :   :           :   :           :   :
  └─(OBJ)─apples

He also likes oranges.    Sentence3:
                           Word3-1[He]  Word3-2[also]  Word3-3[likes]  Word3-4[oranges]
 likes                        POS:PN         POS:ADV         POS:V           POS:N
  ├─(SUBJ)─He               BASE:he        BASE:also       BASE:like       BASE:orange
  ├─also                      :   :          :   :           :   :           :   :
  └─(OBJ)─oranges
```

図 1: 文脈モデル

3 文脈処理を取り入れた機械翻訳

前節で示した文脈処理の機械翻訳における有効性を，この手法を導入した英日機械翻訳システム Shalt2[7] による実際の翻訳例 (図 2, 図 3) を用いて示す．図 2では比較のために文脈情報を用いずに単文内の情報のみで翻訳した結果を併記し，図 3では文脈情報を用いた翻訳結果のみを示した．

3.1 also, only 等の副詞の修飾範囲の決定

図 2は図 1と同じ文章の翻訳結果である．この文章中の文 (2) 及び文 (3) では，副詞 also の修飾範囲 (係り先) が曖昧であり，この曖昧性は単文内の情報のみでは解消できない．一般的にこのような副詞は，前方にある文内容と対比させて修飾範囲を強調する形で用いられることが多く，そのような場合には対比内容を含む文を判別し，対比部分を認識することで修飾範囲が決定できる．

文 (2) を例に取ると，文脈モデル中の前方の文の構造を参照し，also を取り巻く語 {Tom, like, apple} のうち like と apple が文 (1) に存在することから，文 (1) を対比文と判断した上で，文 (2) で John と入れ替わっている語 Tom を also の係り先と判断している．その結果，単文翻訳においては，also が (単文内の情報のみでは修飾範囲が決定できないため) デフォルトで述語に係るものとして翻訳されているが，文脈翻訳においては also は Tom を修飾するものとして翻訳されている．

(1) John likes apples.
＜単文翻訳＞ ジョンは、リンゴを好みます。
＜文脈翻訳＞ ジョンは、リンゴを好みます。

(2) Tom also likes apples.
＜単文翻訳＞ トムは、リンゴを、同様に好みます。
＜文脈翻訳＞ トムも、リンゴを好みます。

```
likes
 ├─(SUBJ)─Tom
 │          └─also
 └─(OBJ)─apples
```

(3) He also likes oranges.
＜単文翻訳＞ 彼は、オレンジを、同様に望みます。
＜文脈翻訳＞ 彼は、オレンジも好みます。

```
likes
 ├─(SUBJ)─He [←Tom]
 └─(OBJ)─oranges
            └─also
```

図 2: 文脈翻訳例 (I)

文 (3) でも同様にして，文 (2) を対比文と判断し，apple と入れ替わっている語 orange を also の係り先と判断して翻訳している．

3.2 代名詞の照応先決定

代名詞の照応先を決定する処理は，文脈の参照を必要とする処理の代表的なものである．図 2の文章の文 (3)

では, 依存構造中で示されている通り, 代名詞の照応先が正しく解析されているが, 代名詞は敢えて代名詞として翻訳文を生成している. 翻訳においては, 大抵の場合, 原言語だけでなく対象言語にも代名詞に対応する語が存在するため, 照応先が100%の精度で決定できない限り, 代名詞を敢えて照応内容に置き換えて訳すよりも代名詞としての曖昧性をそのまま対象言語に残す形で翻訳文を生成することで, 誤った照応先を選択した場合の危険性を避けるためである. 但し, 代名詞の照応先を決定しておくことで,

- 代名詞を格内容に持つ述語の語義 (訳語) 選択の情報が得られる
- 代名詞と照応先の位置関係が翻訳文中で逆転する場合に, 代名詞を照応先の語と置き換えられる
  (例えば英語と日本語では, 主節と従属節の位置関係が逆転する場合がある)
- 代名詞の訳語決定の情報が得られる
  (例えば英語の they は, 照応内容が物か人かによって「それら」「彼ら」のように訳し分ける必要がある)

という利点がある. 例えば, 以下のような英文を翻訳する際には, 代名詞の照応先決定が有効である.

> I will eat the cake, if you don't take it.
> もし君がそのケーキを食べないなら、僕が (そのケーキを) 食べるよ。

Shalt2 には翻訳文における代名詞と照応先の位置的な逆転を扱う機能を組み込んでいないため, 上記の翻訳は Shalt2 によるものではないが, 図 2の文章中の文 (3) において単文翻訳と文脈翻訳で述語 like の訳語が異なっているのは述語の訳語選択において格内容の照応先を参照しているためである. [2]

3.3 文脈内での一貫性保持による曖昧性解消

図 3は, 市販されている英文計算機マニュアルから抽出した文章の翻訳結果である. この文章では, 文 (2) の … know the flow of a job … における前置詞句 of a job, 及び文 (11) の … places the job on a job queue. における前置詞句 on a job queue は係り先が曖昧であるが, 文脈中の他の文を参照すると各々, 文 (7) の The flow of a job … で flow に, 文 (15) の … is placed on an output

[2]John と Tom は, この翻訳処理において用いた辞書の中では人名として登録されていないため, he と異なり「人間」の意味素を持たないので, like の訳語が異なっている.

(1) Tracking Your Job
ユーザーのジョブを追跡すること

(2) It is important to know the flow of a job so that you can track it through the system and display or change its status.
ユーザーが、システムを通して、それを追跡できて、およびその状況を表示できるか、あるいは変更可能なように、ジョブの流れを知っていることは重要です。

(3) This allows you to:
これは、ユーザーにとって、以下を行なうことを可能にします。

(4) End or hold a batch job.
バッチ・ジョブを終了することあるいは保持すること

(5) Answer messages sent by the system.
システムによって送られるメッセージに答えること

(6) Control printer output.
印刷装置の出力を制御すること

(7) The flow of a job can have up to five steps:
ジョブの流れに、最大5のステップがあり得ます:

(8) 1.
1

(9) A user or program submits a job to be run.
ユーザーあるいはプログラムは、実行されるためのジョブを実行依頼します。

(10) 2.
2

(11) The system places the job on a job queue.
システムは、ジョブ待ち行列に、ジョブを置きます。

(12) 3.
3

(13) The system takes the job from the job queue and runs it.
システムは、ジョブ待ち行列から、ジョブを取り、それを実行します。

(14) 4.
4

(15) If this job creates some information (output) that needs to be printed, the printer output is placed on an output queue.
このジョブが、印刷される必要があるいくつかの情報 (出力) を作成する場合には、印刷装置の出力は、出力待ち行列に配置されます。

(16) 5.
5

(17) The system takes printer output from the output queue and sends it to the desired printer to be printed.
システムは、出力待ち行列から、印刷装置の出力を取り込み、印刷されるための必要な印刷装置に、それを送ります。

図 3: 文脈翻訳例 (II)

queue. で place に確定的に係っていることから，文脈中では同じ語が同じような語に係るようにするというヒューリスティックスを適用して，各々の係り先を決定することができる．このように，同じ語基を取る語について係り先の語や語義の決定に関して文脈内で一貫性を保つように処理することで，曖昧性解消の精度 [3] や構文解析に失敗してしまった文法的不適格文の解析精度 [6] を向上させることができる．

3.4 省略の補完

図 3の文章における文 (3) のように，単文で完結せずに後続の文内容を示唆するコロンで終了している場合，文脈モデル中の後続の文の情報を参照することで do the following や the following などの概念を補って翻訳することが可能になる．また，同じコロンで終了していても，文 (7) では省略を含んでいるとは判断されない．

3.5 叙法の判別

図 3の文章における文 (3)～文 (6) のような文は，単独では，この場合のように箇条書きなどで主語の省略されているパタンなのか，命令文なのかが判断できない．このような場合には，文脈モデル中の前方の文を参照することにより，叙法を判断することができる．

4 おわりに

以上，単純な文脈モデルを用いて処理の単位を単文から文脈に拡張した自然言語処理手法を提案し，機械翻訳におけるその有効性を示した．入力文章中の各文の構文解析結果を蓄積した程度の単純な文脈モデルであっても，多義性や係り受けの曖昧性の解消に関する情報を同文脈中の各文で相互に参照し補完し合うことで曖昧性解消の精度を向上させたり，前後の文の情報を参照することで省略の補完，代名詞の照応先の決定などが可能になるなど，単文単位の処理に較べ，質的な向上を図ることができた．

本手法は，あくまで表層的な情報で対処可能な問題のみを処理し，全体的な解析精度を少しでも向上させることを目的としており，完全な解析を指向するものではない．そのため従来の文脈処理に対する大きな期待に応えるほどの精度向上はもたらさない反面，文脈処理用の特別な知識に依存しないことから極めて頑健性で実用性が高い．

また，従来の文単位の処理と比較すると，文脈モデルを構築し参照する処理において計算量が増加しているが，機械翻訳過程全体で見ると，文脈処理の導入による解析精度の向上が，後続の変換生成処理の負荷を下げるため，全体の処理時間はほとんど変わらない．特に，本手法の有効が高い技術文書において，文脈の大きさが 100 文から 200 文程度の場合には，従来の単文単位の処理よりも処理時間が短縮されるケースも確認されている．

参考文献

[1] 長尾真，辻井潤一，田中一敏：”意味及び文脈情報を用いた日本語文の解析－文脈を考慮した処理”，情報処理，Vol. 17, No. 1, pp.19-28, (1976).

[2] Isahara,H. and Ishizaki,S.: ”Context Analysis System for Japanese Text”, In Proceedings of COLING-86 (1986).

[3] 那須川哲哉：”文脈制約を利用した曖昧性解消”，第 7 回人工知能学会全国大会 (1993).

[4] 那須川哲哉：”文脈制約と文脈選好を利用した文脈処理システムＤＩＡＮＡ”，情報処理学会自然言語処理研究会 NL98-8 (1993).

[5] 那須川哲哉：”自然言語解析における複数文一括処理手法”，第 8 回人工知能学会全国大会 (1994).

[6] 那須川哲哉：”文脈情報を利用した不適格文の構文解析”，第 50 回情報処理学会全国大会 (1995).

[7] Takeda,K., Uramoto,N., Nasukawa,T. and Tsutsumi,T.: ”Shalt2 - A Symmetric Machine Translation System with Conceptual Transfer”, In Proceedings of COLING-92 (1992).