可搬的知識を利用した機械翻訳

6Q－6

武田 浩一
日本アイ・ビー・エム株式会社 東京基礎研究所

1 まえがき

商用機械翻訳システムの普及とともに、多様な分野および言語対での文書の翻訳が可能になってきた。特定の言語対に対して、1つの機械翻訳システムを利用していく場合に、ユーザにとっての最大の問題は、学習機能の不足により、同じ種類の誤りを何度も繰り返し訂正しなければならないことと、異なる分野に対する、ユーザ辞書や学習知識をどう管理するかの2点であろう。

既存の機械翻訳システムの多くは、辞書項目の更新機能と、単純な訳語選択の優先度の記憶機能をユーザに解放しているが、レキシコンの記述能力（いわゆる強力な lexical semantics[6] による自然言語処理を可能にするもの）があまり高くないため、語義あるいは係り受けの選択基準をユーザが十分に表現できない。また、一般的なユーザがこのような精密な情報を含んだレキシコンを作成することも困難である。現時点では、機械翻訳システムに文脈や意味理解を含めた深い処理を要求するのは無理といえるので、用例の効果的な利用 [7, 4, 8] や、対話的な学習 [3, 1] により習得できる知識の拡張が最も有望である。

多分野の翻訳を行なう場合には、上記の用例や学習知識が複数分野の知識を反映することになるため、よほど強力な分野の特定能力がない限り適当でない用例や学習知識を翻訳に適用することが避けられず、一元的な知識の管理には問題がある。従って、本論文では可搬的知識という概念を提案し、翻訳された文書（これは同時に用例になり得る）と、その文書をユーザが後編集などにより修正した場合にシステムが学習できる知識を対にして管理し、機械翻訳の最も基本的な3つの種類の曖昧性の解消にこれらを利用する手法について述べる。

2 可搬的知識

可搬的知識（以後、PKS と記す）は、次のような3種類の曖昧性解消のための優先度を表現する知識とする。

(PK1) 語義
(PK2) 句・節の係り受け
(PK3) 訳語の選択

Portable Knowledge Sources for Machine Translation
Koichi Takeda
Tokyo Research Laboratory, IBM Research

このような知識はユーザが直接記述するか、あるいは後編集において翻訳文書を訂正する操作を通してシステムに記憶される。

PK1 の知識の例として、“Delete the *line*.” という文における、語 line が語義1「行」を意味し、この分野（あるいは文書）での語義は、ほぼこれに限られるというときには、

```
(PK1 (``line'' (cat n)) (sense 1))
```

と表現する。同様に、係り受けの知識 PK2 は、“Order the publication through the IBM branch *serving your locality*.” といった現在分詞の係り先が “branch” であることを

```
(PK2 (``serve'' (cat v) (form prsprt))
     ADJP (``branch'' (cat n)))
```

と表現する。PK3 の例としては、“memory chip” の訳語が(1)「メモリー・チップ」と(2)「記憶素子」のように複数個あり、(2) を優先するという場合に、

```
(PK3 (``memory chip'' (cat n))
     (``記憶素子'' (cat n)))
```

と表現する。使用する機械翻訳システムの特性に応じて、品詞・素性情報や各種の意味情報を上記の PKS に加えてよい。1つの文書に対して作成された PKS は、その生成時刻が新しいほど優先されるものとする。

3 可搬的知識の動的な構成

PKS の集合は、各文書に対応づけられ分散している。新たに文書を翻訳する時は、既に存在している文書のうち、特に関連のある文書や同一分野の文書のリストをユーザが指定することで、参照すべき PKS が決定される。これを文書リストと呼ぶ。例えば、技術文書の翻訳などでは、最初に用語集や索引を翻訳し、訳語を統一した後に第1章から順に翻訳するのが効果的である。このような順序をもとに参考文書のリストが与えられなくとも、1つの文書に現れる頻出語を比較して、ある程度関連した文書を自動的に推定することも可能である。得られた文書のリストに対応して、翻訳に利用される PKS の集合が優先度の低い順に並べられたリストを得ることができる。

ユーザは、ある文書の翻訳中に、このような文書リストから文書を削除したり、文書を追加したりすることで、柔軟にPKSを変化させて最適な翻訳知識に調整することができる。PKSそのものを破壊的に書き換えるのではないため、複数ユーザが1つの機械翻訳システムを共有する場合でも、各ユーザに都合のよい翻訳知識を重複なしに管理できる。一見しただけでは類似した分野でも、訳語や語義が異なることは多く、文書単位にこのような翻訳知識を対応づけることで、局所的に語義の一意性[2, 5]や係り受けの一意性を満たすような分野の細分化と、そのような単位で成り立つ翻訳知識の管理が可能になる。

文書リストで与えられたPKSに対して、次のような方法で翻訳知識としての妥当性の程度を計算することができる。

1. 同じ種類のPKS毎に、有向グラフを作る。
   - (PK1 語w 語義j) に対して、ノードwからノードjへの有向枝をもつグラフ。
   - (PK2 語x 関係r 語y) に対して、ノードxからノードyへの、ラベルrの有向枝をもつグラフ。
   - (PK3 語s 語t) に対して、ノードsからノードtへの有向枝をもつグラフ。
2. PK1およびPK3の有向グラフにおいて、同一のノードから複数のノードに出ていくような、競合する有向枝の総数をC1とする。
3. PK2 の有向グラフにおいて、3つのノードn1,n2,n3があり、n1からはn2およびn3への有向枝があり、n2からもn3への有向枝があるような、競合する経路の総数をC2とする。

直観的には、C1は文書リスト中のPKSに現れる、語義と訳語選択の曖昧さを示し、C2は係り受けの潜在的な曖昧さを示している。性質の良い文書リストを選べば、C1もC2も0に近付くと期待できる。また、この値は文書リストの要素の追加、削除により単調（非減少、非増加）に変化する。

4 可搬的知識の利用

2節で述べたPKSは、最も単純な情報のみを記述しているので、種類の異なるものとは互いに独立に適用できる。ある文書を翻訳する時、もし、語義や訳語選択の曖昧性解消に適用できるPKSが複数個あるときは、リストでの順序とPKS集合内でのPKSの生成時刻を用いて、最も優先度の高いPKSが一意に定まる。また、該当するPKSが1つも存在しない時は、システムがデフォールトで選択する語義や訳語を用いる。ユーザが、このようにして得られた語義や訳語を後編集で訂正すれば、その知識が翻訳中の文書に対応するPKSとして新たに記憶される。係り受けの解消も、最も優先度の高いPKSを含む経路を優先すれば同様に行なえる。

翻訳された文書が増加するにつれて、PKSが分散された形で大量に存在することになり、ユーザにとり扱いにくくなる。また、別の環境でこのようなPKSを利用したい時には、PKSを1つにまとめてとり出せることが望ましい。このような場合には、分散したPKSをコンパイルし、1つのユーザ辞書を作成することができる。コンパイルは、ユーザが文書リストを指定し、そのリストに含まれるPKSの和を単純にとればよい。PKSの線形順序は(文書id、生成時刻)の情報を保持することにより決定できる。前節で述べた有向グラフを構成し、競合する有向枝や経路のうち、どの選択が最も望ましいかをユーザが指定すれば、それを満たすような文書リスト中の文書の並びが存在するかどうかは容易に決定できる。2節の方法と、この並び換えの方法により、コンパイルすべきPKSの文書の集まりと、その順序がある程度定量的に評価できる。また、一旦コンパイルされたPKSは、もとの文書との対応が失われていないので、その後、リスト中に別の文書を加えたり、削除したとしても、再計算によりユーザ辞書も自動的に更新できる。これは、関係データベースのビューの概念に近いといえる。

参考文献

[1] C. Boitet and H. Blanchon. "Dialogue-Based MT for monolingual authors and the LIDIA project". In *Proc. of the Natural Language Processing Pacific Rim Symposium '93*, pages 208–222, Fukuoka, Japan, Dec. 1993.

[2] W. Gale, K. Church, and D. Yarowsky. "One Sense Per Discourse". In *Proc. of the 4th DARPA Speech and Natural Language Workshop*, 1992.

[3] H. Maruyama, H. Watanabe, and S. Ogino. "An Interactive Japanese Parser for Machine Translation". In *Proc. of the 13th International Conference on Computational Linguistics*, pages 257–262, Helsinki, Aug. 1990.

[4] K. Nagao. "Dependency Analyzer: A Knowledge-Based Approach to Structural Disambiguation". In *Proc. of the 13th International Conference on Computational Linguistics*, pages 282–287, Helsinki, Aug. 1990.

[5] T. Nasukawa. "Discource Constraint in Computer Manuals". In *Proc. of the 5th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 183–194, Kyoto, Japan, July 1993.

[6] J. Pustejovsky. "The Generative Lexicon". *Computational Linguistics*, 17(4):409–441, December 1991.

[7] S. Sato and M. Nagao. "Toward Memory-based Translation". In *Proc. of the 13th International Conference on Computational Linguistics*, pages 247–252, Helsinki, Aug. 1990.

[8] E. Sumita and H. Iida. "Experiments and Prospects of Example-Based Machine Translation". In *Proc. of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 185–192, Berkeley, June 1991.