E-014

句置換に基づく用例ベース翻訳

EBMT based on Phrase Replacement

鈴木　博和†
Hirokazu Suzuki

1. はじめに

近年、大規模コーパスを用いた翻訳方式に関する研究が盛んにおこなわれており、中でも用例の原文・訳文を用いて入力原文に対する訳文を生成し翻訳を行う用例ベース翻訳(Example-based MT, EBMT)は、従来の規則に基づいた翻訳方式(Rule-based MT, RBMT)にあったような辞書開発者の負担や多言語翻訳の開発コストを大幅に削減することが可能であるため、注目されている。

我々は、これまでの翻訳知識が蓄積され信頼性が高いRBMTと上記のような特徴を持つEBMTを融合させたハイブリッド翻訳方式の実現を目指し、開発を行ってきた。商用機械翻訳ソフトウェア[1]では単語の置換に基づくEBMT機能を実現している。これは差分を単語列としてとらえ、その訳語を用例(翻訳メモリ,TM)の単語対応結果に基づいて挿入・削除・置換することにより訳文を合成する手法(単語置換ベースEBMT)である。

しかし、入力文と用例との差分が大きくなると、差分を単語列として処理するよりも、句として扱った方が訳文生成しやすい場合がある。本稿では、用例原文・訳文から句対応を抽出し、入力文と用例原文との差分を考慮して、用例訳文中の句の置換を行うことによって訳文生成を行う、句置換ベースの EBMT の手法について述べる。

2. 句置換ベース EBMT システム

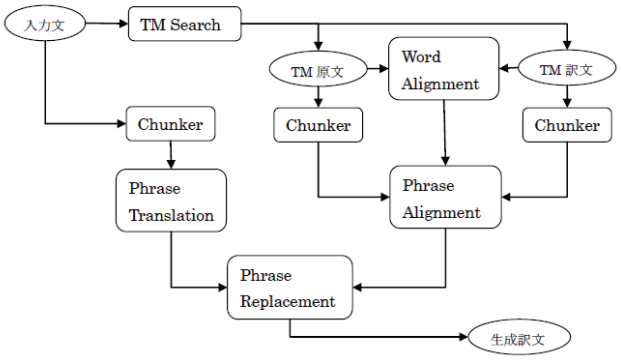

図 1　句置換ベース EBMT システムの概要図

句置換ベース EBMT システムは大きく分けて以下の 6 つ処理から構成されている(図 1)：

(1)**TM 検索**(TM Search)
(2)**単語対応**(Word Alignment)
(3)**句候補抽出**(Chunking)
(4)**句対応判定**(Phrase Alignment)
(5)**句翻訳**(Phrase Translation)
(6)**句置換**(Phrase Replacement)

これらについて順に説明する。

2.1 TM 検索処理

TM から入力文に最も類似した用例を検索する。我々の TM では用例原文・訳文の形態素情報・構文情報、並びに原文・訳文間の単語対応情報などは一切必要としない。従って 2 言語のパラレルコーパスから容易に TM を作成することが可能である。

TM からの類似文検索は表層文字列(日本語の場合)、単語列(英語の場合)の編集距離に基づいて類似度を計算する。従ってカバレージを上げるために類似度を低く設定して検索を行う。ここでは、以下の入力文：

> To reduce the memory access of the above memory means, and to reduce the time required for reading and rewriting registration data in the memory means.

に対して、次の用例が検索されたとする：

> *E:To reduce the memory capacity of a memory means, and to reduce the time required for recording information data in the memory means.*
> *J:記憶手段の記憶容量を低減すると共に、記憶手段に情報データを記録するに要する時間を低減する。*

2.2 単語対応処理

TM 検索により見つかった最も類似度が高い用例の原文・訳文に対して形態素解析処理を行う。次に両者の単語対応を辞書ベースで行う。

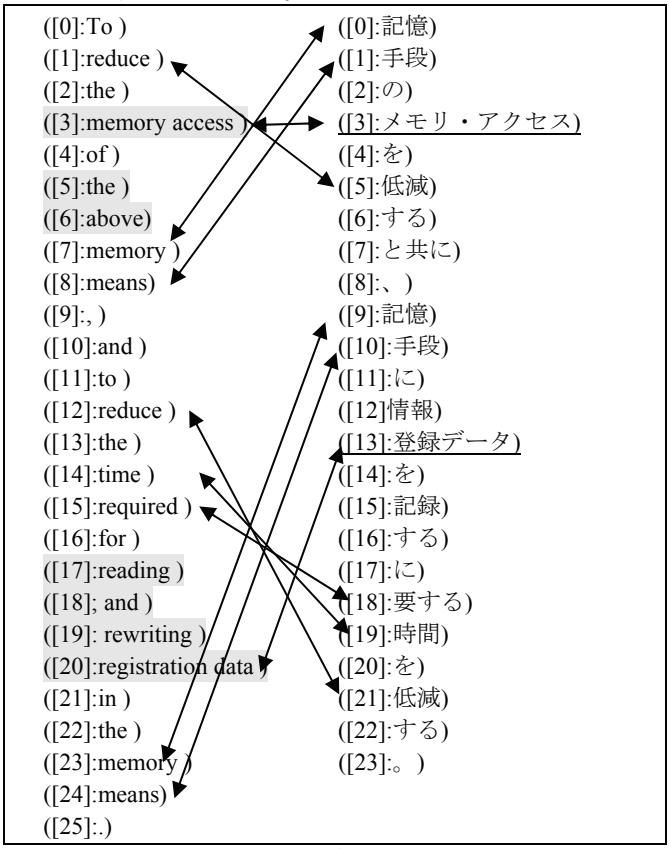

図 2　単語置換 EBMT 結果

†（株）東芝　研究開発センター　知識メディアラボラトリー

まず、辞書中に見出し・訳語のペアとして存在し、単語の対応が一意に同定できるものをアンカーとする。それ以外の単語対応候補をアンカーの近傍順に絞り込んでいくことによって、単語対応を行っていく。

一方、用例の単語対応情報と入力文と用例原文間の差分情報から、差分単語列が、用例に対しての挿入・削除・置換のどれになるかを調べる。それに基づいて用例訳文に対する編集処理を行い、単語レベルでの差分を用例訳文に反映させ一時的な訳文(**一次訳文**)を生成する。この処理は**単語置換ベース EBMT** に相当する(図 2)。

網掛け部分が入力文と用例原文間の差分を表わし、下線部が訳語が置換された部分を表わす。単語置換ベース EBMT では置換の対象とする単語を名詞に限定しているため、その他の単語は置換されていない。

2.3 句候補抽出処理

用例原文、用例訳文、入力文、一次訳文の 4 つに対して構文解析を行い、それぞれから句を抽出する。はじめに各構文解析木のノードを辿っていき、最初の句候補(句アンカー)を見つける。

次にこの句アンカーの先頭・末尾の位置をヒューリスティックな知識に基づいて前後に移動させることによって句を移動・伸縮させ、複数の名詞句候補を生成する。

2.4 句対応判定

2.3 によって生成された複数の句候補から、用例原文・用例訳文間の句対応、入力文・一次訳文間の句対応を見つける。ここで以下の評価式を用いる：

単語数類似度 $(len\_rsmbl) = slen/tlen$
対応候補充足度 $(suff) = (wsnum + wtnum)/(snum + tnum)$
単語対応度 $(crrspnd) = (al\_num)/(snum + tnum)$
用例句対応度 $(pc\_tm) = 2 * suff * crrspnd/(suff + crrspnd)$
入出力句対応度 $(pc\_io) = al\_num$

slen は原文側の句の単語数、tlen は訳文側の句の単語数を表わす。snum,tnum は句の中で 2.2 の単語対応処理によって対応がとれる可能性がある品詞(名詞・動詞・形容詞・副詞など)を持つ単語数を表わす。wsnum,wtnum は snum,tnum に対して、品詞によってカウントの重みを変えたものである。単語置換ベース EBMT で置換の対象とするのは名詞なので、名詞であれば 1、それ以外の品詞の場合は 0.7 としてカウントする。0.7 の値は実験により経験的に定めた。

以下の基準を満たす句同士を対応している句とする：

用例原文・訳文間の句対応：
$0.5 \le len\_rsmbl \le 2.0$ かつ $\max(pc\_tm)$
入力・一次訳文間の句対応：
$0.3 \le len\_rsmbl \le 3.3$ かつ $\max(pc\_io)$

上記において、句対応候補の位置に重複があるもの同士で、句対応度(pc_tm, pc_io)の比較を行う。従って上記で得られる句対応は 1 つとは限らず、複数個所で句対応が得られる可能性がある。また、用例の句対応評価基準は単語数が類似したものを優先するようにし、入力文・一次訳文間の句対応評価基準は単語対応を優先している。これにより入力文と一次訳文とで単語数に大きな隔たりがあって差分が大きいとしても、単語対応がとれていれば句の対応を行うことが可能になっている。al_num が単語対応数を表わす。ここではカウント方法として 2 つ示す：

・対応カウント法A：原文側句候補内の単語の対応先が訳文側句候補内に存在している場合、名詞ならば 1、それ以外の品詞であれば 0.7 とカウントする。
・対応カウント法B：手法Aの値を 2 倍する。原文側句候補内の単語の対応先が訳文側句候補内に存在していない場合、名詞ならば 1、それ以外の品詞であれば 0.7 をカウントから減算する。訳文側から原文側に対しても同様。

後者は、対応なしをペナルティとして扱っている。

2.5 句翻訳、句置換処理

2.4 で句対応が得られると、まず入力文中の句に対して翻訳を行い名詞句として訳出する(句翻訳)。また入力文中の句に対応した一次訳文側の句を求め、置換すべき範囲を決定する。用例原文・訳文間の句対応で、入力文・一次訳文間の句対応に相当するものがあり、その中に入力文・用例原文間の差分が存在している場合、一次訳文側の句を句翻訳結果と置換する(句置換)。

■用例原文・訳文句対応
[1]the memory capacity of a memory ⇔記憶手段の記憶容量
[2]the time required for recording information data in the memory means⇔記憶手段に情報データを記録するに要する時間
■入力文・一次出力文句対応*(句翻訳結果)*
[1]the memory access of the above memory means⇔記憶手段のメモリ・アクセス*(上記のメモリ手段のメモリ・アクセス)*
[2]the time required for reading and rewriting registration data in the memory⇔記憶手段に情報登録データを記録するに要する時間*(メモリ中の登録データを読み書き直すことに必要な時間)*
■合成訳文(太字はもともとの用例中にあった文字列)
*上記のメモリ手段のメモリ・アクセス*を低減すると共に、*メモリ中の登録データを読み書き直すことに必要な時間*を低減する。

3. 実験結果と今後の課題

特許分野の用例を用いてその中の 429 用例について、句対応精度を検証した。対応カウント手法において手法 A を用いた場合、対応数は 285 でその中で正しい対応数は 210(73.7%)であった。一方手法 B を用いた場合、対応数は 280、正しい対応数は 212(75.7%)であった。句対応においては、対応がとれなかった単語も考慮した方が良いことが分かる。そこで手法 B を用いて、最終的なシステムの生成訳文の品質を評価すると、入力文 332 文について、255 文の合成訳文が生成されそのうちの 62.0%は訳文生成が成功していた。本稿での EBMT 手法においては句候補生成、句対応が極めて重要である。今後はこれらの精度向上を行い、合成訳文の品質向上を行っていく。

参考

[1]The 翻訳プレミアム 2009 プレスリリース, http://www.toshiba-sol.co.jp/news/detail/090119.htm
[2]Nagao, M. 1984. “A Framework of a Mechanical Translation between Japanese and English by Analogy Principle”,Artificial and Human Intelligence,pp.173-180
[3]Carl, M and Way, A. “RECENT ADVANCES IN EXAMPLE-BASED MACHINE TRANSLATION”, Kluwer Academic Publishers