3E−1

日英文対応データの自動付与方法

井ノ上直己　野垣内出

KDD研究所

1. はじめに

現在までに、多数の機械翻訳システムが実用化されてきたが、これらの機械翻訳システムは不完全であり多数の問題を抱えている。そのため、最近では「実例に基づく翻訳」といった新しいパラダイムも起こり、実際の翻訳例を参考にすることの重要性が指摘されている[1]。また、単語を意味的に分類する場合には1つの言語に閉じて行なうよりも、別の言語を考慮した方が効果的であることも示されている[2]。

そこで、2言語間の対応情報を付与した大規模コーパスの構築が求められ、現在までいくつか構築されている[3][4]。しかし、これらのコーパスでは文節あるいは単語間の対応、単語依存構造間の対応を付与しており、生テキストに何らかの処理を施した上で作成されている。そのため、利用者が容易に対応情報の付与されたコーパスを構築することができない。

翻訳援助システムや辞書作成援助システムとしては文字列のままの対応データでも有用であると考えられるため、本稿では、生テキストに対して英語と日本語の文毎の対応を自動的に与える手法について述べる。また、UNIXのオンラインマニュアル約1000文に対して実験を行ない95.6%の正解率を得たので報告する。

2. 英語文と日本語文の対応データ

本稿で提案する手法の目的は、図1に示す英語と日本語の生テキストを入力し、図2に示すように英語文と日本語文の対応を自動的に取ることである。

本稿では、容易に英語と日本語とで対応するテキストが入手できるUNIXオンラインマニュアルを対象とした。

For each filename which is a directory, ls lists the contents of the directory; for each filename which is a file, ls repeats its name and any other information requested. By default, the output is sorted alphabetically. When no argument is given, the current directory is listed. When several arguments are given, the arguments are first sorted appropriately, but file arguments are processed before directories and their contents.

引数 filename としてディレクトリを指定すると、ls はそのディレクトリの内容を出力します。引数でファイル名を指定すると、ls はその名前と要求された情報を出力します。通常、出力はアルファベット順にソートされます。引数を指定しないと、現ディレクトリの内容を出力します。複数の引数を指定すると、引数がソートされますが、ファイル名指定が、ディレクトリ名指定よりも先に処理されます。

図1　生テキストの例

For each filename which is a directory, ls lists the contents of the directory;
引数 filename としてディレクトリを指定すると、ls はそのディレクトリの内容を出力します。

---

for each filename which is a file, ls repeats its name and any other information requested.
引数でファイル名を指定すると、ls はその名前と要求された情報を出力します。

By default, the output is sorted alphabetically.
通常、出力はアルファベット順にソートされます。

When no argument is given, the current directory is listed.
引数を指定しないと、現ディレクトリの内容を出力します。

When several arguments are given, the arguments are first sorted appropriately, but file arguments are processed before directories and their contents.
複数の引数を指定すると、引数がソートされますが、ファイル名指定が、ディレクトリ名指定よりも先に処理されます。

図2　文対応データ

英語文と日本語文との対応関係の特徴を分析するため、任意に抽出したテキストに対して事前に人手で文毎の対応付けを行なった。その結果を表1に示す。

ここで、Exは英語文の数がxであることを、Jyは日本語文の数がyであることを示す。つまり、表中の678は英語1文に対し日本語1文が対応した数を示し、49は英語1文に対し日本語2文が対応した数を示す。表1から約86%が1対1対応し、実線で囲んだ範囲で約98%をカバーしているのが分かる。また、英語に対応する日本語が存在しない場合も4例あった。

表1　対応関係

	J0	J1	J2	J3	J4	J5	J6
E0	-	0	0	0	0	0	0
E1	4	678	49	12	2	1	1
E2	0	24	13	3	0	0	0
E3	0	3	0	0	0	0	0
E4	0	1	0	1	0	0	0
E5	0	0	0	1	0	0	0

An Automatic Alignment Method of Japanese and English Sentences
Naomi INOUE, Izuru NOGAITO
KDD R & D, *Asahi Chemical Ind. Co.,Ltd.

3. 処理アルゴリズム

図1に示す英語と日本語の生テキストから図2に示す対応データを自動的に得るためには、生テキストを (1) 文に分割する処理、(2) 分割された文に対して英語と日本語で対応を取る処理が必要となる。しかし、ここでは本手法の主処理である (2) について述べる。

英語と日本語とで対応を取るアルゴリズムを以下に簡単な例を用いて説明する[5]。今、長さ3の記号列 ($a_1a_2a_3$) と長さ4の記号列 ($b_1b_2b_3b_4$) が存在する時、各記号間で対応を取ることとは図3に示すような格子状のマトリックスを考え、AからBまで格子点を結んだパスを求めることだと考えられる。例えば、図3のパスIは (a_1,b_1b_2)、(a_2,b_3)、(a_3,b_4) という対応を、パスIIは (a_1,b_1)、(a_2,b_2)、(a_3,b_3)、(,b_4) という対応を示す。ここで、(a_1,b_1b_2) とは a_1が b_1と b_2に対応することを表し、(,b_4) は b4 に対応する記号が存在しないことを表す。

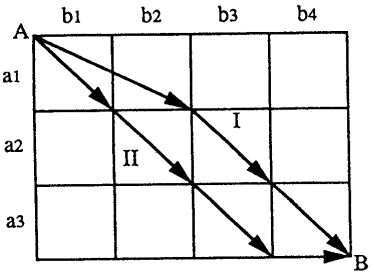

図3　対応処理の例

AからBまでのパスを1つ求めるためには、各格子点を結ぶ際にその尤度を計算し、最終的にBまで到達するパスで最も尤度の高いパスを選択すればよい。

また、前章で示した通り英語と日本語の対応は1対1、1対2、1対3、2対1、2対2、3対1対応で約98%をカバーしている。さらに対応する英語あるいは日本語が無い場合を考慮して以下に示すアルゴリズムを用いた。

```
S(0,0) = 0;
for i = 1 to m
{   for j = 0 to n
    {   S1 = S(i,j-1) + s(a_i,b_{j-1};0,1);
        S2 = S(i-1,j) + s(a_{i-1},b_j;1,0);
        S3 = S(i-1,j-1) + s(a_{i-1},b_{j-1};1,1);
        S4 = S(i-1,j-2) + s(a_{i-1},b_{j-2};1,2);
        S5 = S(i-1,j-3) + s(a_{i-1},b_{j-3};1,3);
        S6 = S(i-2,j-1) + s(a_{i-2},b_{j-1};2,1);
        S7 = S(i-2,j-2) + s(a_{i-2},b_{j-2};2,2);
        S8 = S(i-3,j-1) + s(a_{i-3},b_{j-1};3,1);
        S(i,j) = max(S1,S2,S3,S4,S5,S6,S7,S8); }}
```

上記のS(m,n)が最も高い尤度であり、その尤度を得た時のパスが求める対応である。ここで、$s(a_i,b_j;p,q)$ とは a_iから p 個の文 ($a_i,a_{i+1},...,a_{i+p-1}$) と b_jから q 個の文 ($b_j,b_{j+1},...,b_{j+q-1}$) が対応する時の尤度を表す。

4. 尤度

前章のアルゴリズムで用いている尤度 $s(a_i,b_j;p,q)$ は以下の式で求める。

$$s(a_i,b_j;p,q) = \alpha\frac{1}{\sqrt{2\pi\sigma^2}}exp\{-(r-\mu)^2/2\sigma^2\} + \beta\frac{c(a_i,b_j;p,q)}{l(a_i;p)+l(b_j;q)-c(a_i,b_j;p,q)}$$

ここで、$l(a_i;p)$ は a_iから p 個の文の文字数の総和であり、$c(a_i,b_j;p,q)$ は a_iから p 個の英語文と b_jから q 個の日本語文の両方に共通して出現した文字の数を表す。また、r は $l(b_j;q)/l(a_i;p)$ であり、英語と日本語の文字数の比を表し、μおよびσ^2はそれぞれ r の平均と分散を表す。αおよびβは重み係数である。

上記式の右辺第1項は対応する英語と日本語の文字数の分布は正規分布に従うと仮定し、その確率を計算している。また、数字や"SunOS" などの固有名詞は英語でも日本語でも同じ文字列が利用されることがある。そのため、右辺第2項はそれらの文字列の含まれている割合が高いほど対応する可能性が高いとして、その割合を求めた。この第2項を用いた点が本方式の特徴である。

5. 対応実験

本稿で提案した尤度の有効性を評価するため、第2章で用いたテキストとは別に得た UNIX のオンラインマニュアル約 1000 文 (英語 955 文、日本語 1004 文) に対して、上記式の右辺第2項がある場合と、無い場合とで比較実験を行なった。その結果を表 2に示す。

表 2より、第2項を用いた方が 6.3%正解率が向上している。なお、この実験で用いた変数の値は$\mu = 0.749$、$\sigma^2 = 0.631$ であり第2章で示した対応データから得た。また、$\alpha = 0.25$、$\beta = 1.0$ とした。

表 2　正解率

第2項有り	95.6%
第2項無し	89.3%

6. まとめ

本稿では、英語文と日本語文との対応データを自動的に得る手法について述べ、UNIX のオンラインマニュアル約1000文に対して実験を行なった。その結果、正解率 95.6%を得た。本手法では、言語的な情報（辞書）を用いず統計的に処理できるため、容易に文対応情報を付与したコーパスが構築できる。

今後は、作成したコーパスに検索プログラムを追加し、翻訳辞書作成援助ツールとして利用していきたい。

参考文献

[1] 佐藤理史：「実例に基づく翻訳」，情報処理，Vol.33, No.6, pp.673-681(1992)

[2] Inoue, N.: *"Automatic Noun Classification by Using Japanese-English Word Pairs"*, Proceedings of the 29th Annual Meeting of ACL, pp.201-208(1991)

[3] 江原、小倉他：「電話またはキーボードを介した対話に基づく対話データベース ADD の構築」，情処論文誌，Vol.33, No.4, pp.448-456(1992)

[4] Sadlar, V., et al.: *"Pilot Implementation of a Bilingual Knowledge Bank"*, Proceedings of COLING-90, pp.449-451(1990)

[5] 田中英一，「構造をもつものの距離と類似度」，情報処理，Vol.31, No.9, pp.1270-1279(1990)