自然言語処理 96－7
（1993．7．9）

# 用例検索による韓日・日韓翻訳支援システム

黄 道三　　長尾 眞　　佐藤 理史*
京都大学工学部 電気工学第二教室
*北陸先端科学技術大学院大学 情報科学研究科

**要旨**

近年機械翻訳システムが本格的に商品化されてきているが、翻訳結果が不完全であるので、翻訳支援システムが使われなくてはならない。ところで、実際の翻訳例文は人間が翻訳する場合にも役に立つという考え方に基づいて、用例を用いた翻訳支援システムが開発されている。これには文字照合検索法と単語照合検索法が使われている。文字照合検索法は文字の種類が多く、また類似語の文字表記に同じ文字が使われる傾向の高い日本語と韓国語において適合していると思われる。しかし、この方法では翻訳例文が入力文字列と同じ文字で書かれていないと検索されないので、大量の翻訳例文データベースが要求されるとともに検索時間の問題も考えなくてはならなくなる。そこで、本稿では検索時間を高速化し、文字列の類似度の計算法を改良した文字照合検索法を示す。また、単語の意味的類似度に基づいて、同じ文字が使われていなくても意味的に似ていれば類似例文として検索する意味照合検索法を示す。最後に韓国語と日本語の文を対象にして実験し、その評価を示す。

# ECTM: An Example-Based Korean to Japanese Translation Aid System

Dosam Hwang　　Makoto Nagao　　Satoshi Sato*
Department of Electrical Engineering, Kyoto University
*School of Information Science, Japan Advanced Institute of Science and Technology, Hokuriku

**Abstract**

Although Machine Translation Systems have been commercially developed, there is still no system that can consistently produce acceptable translations. Translation-aid systems by showing example translations have recently been proposed as a candidate of a translator work station. Systems of this kind utilize two kinds of retrieval methods: character-based and word-based matching. Character-based matching is applicable to languages such as Korean and Japanese, which have many different characters and where the same characters often occur in synonyms. However, this method cannot offer any example sentence similar to the input if no example sentence containing the same characters exists. Furthermore, to obtain good results using this method, one obviously requires a large translation database, thus making retrieval times large. In this paper, we present a retrieval method based on character to decrease retrieval time and to find more similar example sentences. In addition, we present a method that does the matching operation based on a thesaurus, in order to retrieve similar example sentences, even if they have no characters in common with the input. We present and evaluate results of the experiments for Korean and Japanese.

## 1 はじめに

1980年中頃から機械翻訳システムが本格的に商品化されてきている。しかし、意味解析および文脈解析などの解析技術がまだ不完全であるので、多義性のある文や、小説または詩などの文学作品の場合にはほとんど翻訳できない。このような問題を克服するために、参考文献[1]によってアナロジーによる翻訳という翻訳方式が提案された。以後、実際の翻訳例文は人間が翻訳する場合にも非常に役に立つという考え方に基づいて、用例を用いた翻訳支援システムが発表されている。これには、佐藤のCTM[2]のように文字を検索の基本単位とする文字照合検索法と、ETOC[3]と中村のシステム[4]のように単語を検索の基本単位とする単語照合検索法の二種類の検索法が使われている。特に、文字照合検索法は文字の類似性を用いて類似例文を検索するので、文字の種類が多く、また類似語の文字表記に同じ文字を使う傾向の高い日本語と韓国語において非常に適合していると思われる。

したがって、本稿では文字照合検索法に基づいて、

- 2文字インデックスを使った文字検索法を用い、文字列の類似度計算法を改良した文字照合検索法を示す。
- また、分類語彙表[5][6]の分類番号による照合法[*1]を加えることによって、同じ文字が含まれている例文がなくても意味的に似ていれば、類似例文として検索する意味照合検索法を示す。

最後に、韓国語と日本語を対象にして、この照合検索法を評価し、その有効性を示す。

## 2 システムの構成

### 2.1 韓国語の特性

韓国語と日本語では、語順が似ているだけでなく、漢字から伝来された漢字語において非常に似ている。ここでは、文字照合検索法が韓国語にも有効であろうということを見るための韓国語の特性だけを述べる。

1. 表記に用いる文字の種類が多い。
   韓国語を表記するために使われる文字数は、ハングルが約2,350種類で、日本語の約7,000種類よりは少ないが、他のヨーロッパの言語よりはずっと多い。また、漢字が併用して書かれている文献もあり、この場合には約7,200種類になる。しかし、本研究にはハングルだけを対象にした。
2. 類義語には同じ文字が使われる場合が多い。
   韓国語の中で漢字語は日本語の漢字のように類義語には同じ文字が使われる。例えば、「고(KO)」は次のように「사고(思考)」に関する類義語に共通に使われている。
   사고(思考)、고찰(考察)、숙고(熟考)、고안(考案)
3. 形態素解析が難しい。
   単語の間、または句の間にホワイトスペースをおいて分ち書きしているが、それが不規則であることが少なくなく、一般的にそれに従わずに書かれている文が多い。また、多義性を持っている文字または単語が多いので、形態素解析が非常に難しい。

上記のような特性をみると、韓国語の場合にも文字照合検索法が有効であろうと思われる。

### 2.2 システムの構成

2.1に述べた韓国語の特性に基づいて、基本的には文字照合検索法を採用し、4章に述べる意味照合検索法の部分を加え、図1のようにシソーラス、類似例文検索、翻訳例文データベースの三つの部分で構成したECTMシステムを作った。

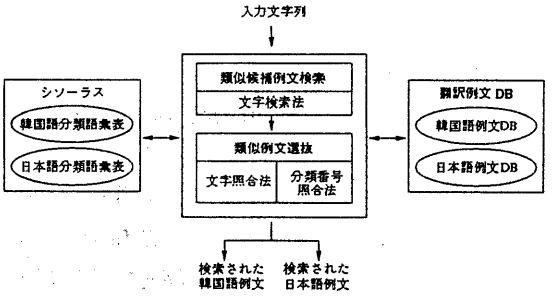



図1: ECTMのシステム構成

翻訳例文データベースはあらかじめ翻訳されている韓国語と日本語との対訳文をデータベースにしたものである。類似例文は類似候補例文検索と類似例文選抜との二つの段階を経て検索される。第一段階の類似候補例文検索は韓国語あるいは日本語が入力文字列として入力されると、文字列の順序を考慮せず、入力文字列に含まれている文字を多く含んでいる翻訳例文を類似候補例文として選抜する部分である。これは大規模の翻訳例文を対象にして検索するので、検索の高速性が求められ、このために3.1章に述べる2文字インデックスと選抜基準値および足切り関数を用いる。第二段階の類似例文選抜では入力文字列と類似候補例文との文字順序も考慮した文字照

[*1] 分類番号照合法と呼ぶ

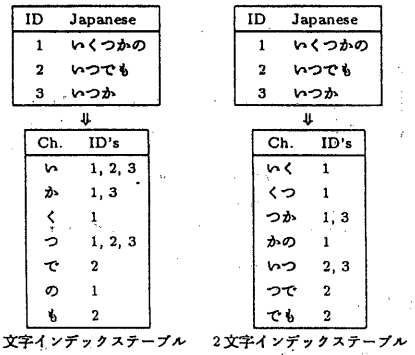



図 2: 文字インデックステーブル

合法と単語の意味的類似度を用いる分類番号照合法を利用して、最終的な類似例文を選ぶ。

## 3 文字照合検索法の改良

### 3.1 文字検索法の改良

CTM では検索の高速化のために 1 文字インデックステーブルと足切り関数を用いた文字検索法を示した。この方法は韓国語のように表記文字の種類の多い言語には非常に有効であると判断される。しかし、1 文字インデックスの場合には、その文字を含んでいる例文が多くなってしまって類似候補例文の数が多くなる可能性が高い。特に、韓国語の文字は日本語の三分の一ぐらいであるので、この現象が韓国語例文の検索には非常に起こりやすい。我々は、検索速度を低下させず、メモリをあまり取らないとすれば、第一段階の類似候補例文検索でも文字列の順序を考慮する方が望ましいと考えた。そこで、本稿では図 2のような 2 文字インデックステーブルを用いた。文字インデックスの長さを大きくすればするほどより似ている例文が検索できるが、逆にテーブルが大きくなって大量のメモリが必要になる。ここでは韓国語単語の平均の長さが約 2 であることを考慮して 2 文字インデックスを用いた。2 文字インデックステーブルとは入力文を左から右へ走査しながら 2 文字ずつ読んでそれをインデックスにし、今走査している文の番号をインデックスの内容にするテーブルである。また、図 3のように各要素に次の要素を指すポインターを置いた一方向リストで構成することによってメモリも効率的に活用した。

文字検索法は、図 2のように表記文字に対して、その各文字が使われている翻訳例文の ID 番号をあらかじめ 2 文字インデックステーブルとして用意する。そしてこの表

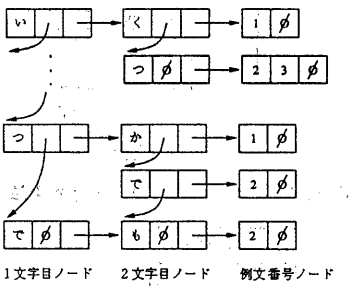



図 3: 2 文字インデックステーブルのデータ構造

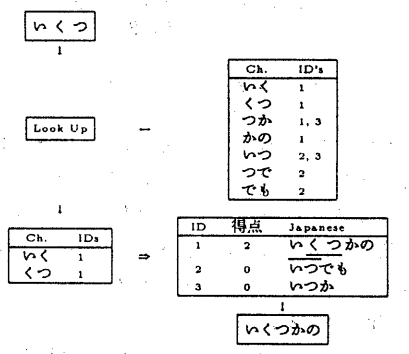



図 4: 2 文字インデックスの文字検索法

より入力文に現れている文字を含んでいる翻訳例文の ID 番号を抽出して、これを多く含んでいる翻訳例文を検索する方法である。図 4の例をあげて説明しよう。

1. 入力文字列に含まれている文字をキーにして、2 文字インデックステーブルよりその文字が使われている翻訳例文の ID 番号を検索する。
2. それぞれの翻訳例文に対して、選抜の得点を計算する。この得点は、検索された文字インデックステーブルに書かれている翻訳例文の ID 番号の個数で求められる。これは、入力文字列と翻訳例文との間で文字の順序を考慮しなかった文字の一致数である。
3. 選抜の得点の高い上位 N(足切り値) 個を選ぶ。

ここで 2 文字インデックステーブルの場合は、図 4のように一つの例文が検索されるが、1 文字インデックステーブルを使うと、図 5のように三つの例文が検索される。

また、検索された翻訳例文に入力文字列に使われている文字がどのぐらい現れているかを検討しないで N をある定数で固定すると、類似性の高い翻訳例文がなくても一定数の類似性の低い文が第 2 段階の文字最適照合に渡されてしまい、第 2 段階に負担を負わせてしまう場合が

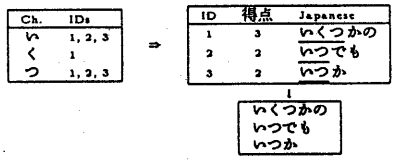



図 5: 1 文字インデックスの文字検索法

```
if (N ≤ 200)
  then if (SL < 6)
         then SV = SL
         else SV = SL * x
  else
```

SL: 入力文字列の長さ SV: 選抜基準値
x: インデックス包含率

図 6: 選抜基準値の計算式

ある。また、第 2 段階での計算時間は第 1 段階より平均約 2 倍かかるので、できるかぎり第 2 段階の計算負担を軽くする方が良い。また、韓国語の表記文字の種類は日本語の約三分の一ぐらいであるので、$N$ の値を上げなくてはならないし、こうすると第 2 段階の負担が増えて検索時間が相当かかる。

したがって、本稿では図 6のような選抜基準値を足切り値に加えて検索速度を上げた。これは入力文字列の長さ*2が 6*3より小さい場合には、入力文字列のすべての文字が含まれている翻訳例文だけを、6 以上の場合には、入力文字列の最小限 $x$% *4以上のインデックスが含まれている文だけを類似候補例文として選ぶということである。

### 3.2 文字照合法の改良

3.1では文字の順序を考慮せず、同じ文字が含まれている翻訳例文を類似候補例文として選んだので、この例文の中にはまだ入力文字列と似ていない文が存在している可能性がある。そして、CTM では類似候補例文から最適類似例文を取り出すために、文字の順序と文字の連続性とを考慮して、一致文字数に基づいて図 7のような文字列類似度を用いた。

しかし、この式では、図 8のように「いくつの観点から考察を行なう」$(I)$ という入力文字列に対して、「3 つの他の観点から考える。」$(E)$ という翻訳例文が検索された場合、文字列類似度を計算する途中、文字「の」$(I_i)$ が「3 つの」$(F)$ と「の観点」$(R)$ との二ヶ所で重ねて

*2 入力文字列に使われている文字の数

*3 これはヒューリスティックな数値である。

*4 デフォルトは 70% で、ユーザ定義数値である。

$$S(A,B) = s(x,y)$$

$$s(i,j) = \begin{cases} 0 & \text{if } (i=0) \vee (j=0) \\ max \begin{pmatrix} s(i-1,j-1) + min(cm(i,j), W^*) \\ s(i-1,j), \\ s(i,j-1) \end{pmatrix} & if (1 \le i \le x) \wedge (1 \le j \le y) \end{cases}$$

$$cm(i,j) = \begin{cases} 0 & \text{if } (i=0) \vee (j=0) \\ cm(i-1,j-1) + m(i,j) & if (1 \le i \le x) \wedge (1 \le j \le y) \end{cases}$$

$$m(i,j) = \begin{cases} 1 & if a_i = y_j \\ 0 & if a_i \neq b_j \end{cases}$$

*W(連続値)=4

図 7: CTM での文字列類似度の計算式

| | | つ | の | 他 | の | 観 | 点 | か | ら | 考 | え | る |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| い | . | . | . | . | . | . | . | . | . | . | . | . |
| く | . | . | . | . | . | . | . | . | . | . | . | . |
| つ | . | 1 | . | . | . | . | . | . | . | . | . | . |
| の | . | . | 2 | . | 1 | . | . | . | . | . | . | . |
| 観 | . | . | . | . | . | 2 | . | . | . | . | . | . |
| 点 | . | . | . | . | . | . | 3 | . | . | . | . | . |
| か | . | . | . | . | . | . | . | 4 | . | . | . | . |
| ら | . | . | . | . | . | . | . | . | 4 | . | . | . |
| 考 | . | . | . | . | . | . | . | . | . | 4 | . | . |
| 察 | . | . | . | . | . | . | . | . | . | . | . | . |
| を | . | . | . | . | . | . | . | . | . | . | . | . |
| 行 | . | . | . | . | . | . | . | . | . | . | . | . |
| な | . | . | . | . | . | . | . | . | . | . | . | . |
| う | . | . | . | . | . | . | . | . | . | . | . | . |

図 8: CTM での文字列類似度の計算例

計算される。

この場合、本稿では図 9のように後ろの単語だけに対して計算が行なわれるようにして、さらに妥当な類似度が求められるようにした。

## 4 意味照合検索法

3章に述べた方法では、入力文字列と同じ文字が含まれていないと、類似例文として検索されないので、大規模の翻訳例文データベースが必要になる。したがって、本稿では CTM の基本趣旨とは少しずれてしまうが、単語の意味的類似度を用いて類似例文を検索する意味照合検索法を考えた。この方法では、まず文字検索法を用いて類似候補例文を検索して、文字照合法と分類番号照合法とを用いて類似例文を選抜する。

```
if (R1 ≡ Fn)
  then SC = SC − SC(Ii, Fn)
  else
```

ただし、

$F : E[j, \cdots, k] = I[m, \cdots, n]$ $R : E[k+x, \cdots, l] = I[n, \cdots, o]$

I: 入力文字列 E: 例文文字列 SC: 文字列類似度

図 9: 文字列類似度の計算に付加された式

$$SA = \sum SAW_i / \sqrt{IN} * \sqrt{\frac{MN}{IN}}$$
$$SAW = \{k | I[1,\ldots,k] = R[1,\ldots,k] \wedge I[k+1] \neq R[k+1]\}$$

$SA$：文の意味的類似度　$SAW$：単語の意味的類似度

$IN$：分類番号照合の対象数　$MN$：分類番号が一致された単語数

図 10: 意味的類似度の計算式

$$TA = CA + SA$$

$TA$：類似度　$CA$：文字列類似度　$SA$：意味的類似度

図 11: 類似度計算式

例えば、分類番号照合法で単語の意味的類似度を計算する部分を「$」記号で囲んで入力すると、次のような手順で類似例文が照合される。

1. 入力文字列の中で「$」記号で囲まれていない部分 (文字照合部分と呼ぶ) を対象にして文字検索法を用いて第1次類似候補例文を検索する。
2. 文字照合部分に対して文字照合法で文字列の類似度を求めて、第2次類似候補例文を選ぶ。
3. 「$」記号で囲まれている部分を対象にして、分類語彙表から分類番号を検索し、意味的類似度を計算して、最後に類似例文を選ぶ。

入力文字列の中で意味的類似度で照合したい部分 (分類番号照合の対象と呼ぶ) については、分類語彙表の分類番号に基づいてその単語の意味的類似度を求めて類似例文を照合することにした。単語の意味的類似度の計算には図10のような式を用いた。

また、分類番号照合法には次のような三つの表記方式を導入した。ここにSは文字列を表す。

1. 完全一致: $S$
2. 左最長一致法: $S * $
3. 右最長一致法: $ * S$

最終的な類似度は図11のように「$」で囲まれていない文字列類似度と「$」で囲まれている単語の意味的類似度とを合わせることによって求めた。この例を図12に示す。

ここで、図13のように照合の対象単語が用言の場合には、その単語の前に「$v」と書き、用言の原形に変換した後、分類語彙表より分類番号を引くようにした。

## 5 翻訳例文データベース

翻訳例文データベースは韓国語例文とそれに対訳づけされている日本語例文の翻訳例文ファイルと、これを効率的に検索・管理するためのインデックスファイル、管理ファイルとで構成されている。本システムでは、大量

入力文: $*私$は　$彼$に　$*本$を　$v貸して$もらった。
検索文: 昨日 彼女は　先生に 卒業 論文を　貸していただいた。

- $SAW_1 = 5$
  - $I_1$:　私 12000 1 20
  - $R_1$:　彼女 12000 3 20
- $SAW_2 = 3$
  - $I_2$:　彼 12000 3 10
  - $R_2$:　先生 12020 6 115
- $SAW_3 = 3$
  - $I_3$:　本 13160 1 30
  - $R_3$:　論文 13154 6 60
- $SAW_4 = 7$
  - $I_3$:　貸す 23770 3 100
  - $R_3$:　貸す 23770 3 100

$SAW = 5 + 3 + 3 + 7 = 18$

$IN = 4,\ MN = 4$

$SA = 18/\sqrt{4} * \sqrt{\frac{4}{4}} = 9$

図 12: 意味的類似度の計算例

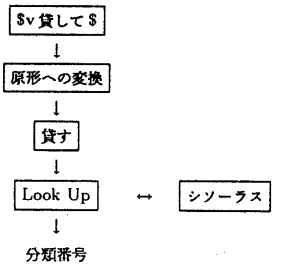



図 13: 用言の分類番号照合法

の翻訳例文が必要であり、この例文が多量のメモリを占めるので、検索時間を短縮するとともに、メモリを効率的に活用するために、例文ファイル構成の最適化を考えなければならない。そこで、インデックスファイルを3.1に述べたように各要素に次の要素を指すポインターを置いた一方向リストで構成した。また、いろいろな分野の例文を扱わなければならないので、検索される例文が書かれている文献名や位置などの情報を提供するために、例文ファイルの管理も考慮しなければならない。そして、図14のように同じ分野の複数のファイルは一つのディレクトリの下に置き、さらにこれらの複数のディレクトリが集まって翻訳例文データベースとなる階層的構造に構築することにした。したがって、プログラムの実行時に図15のような翻訳例文データベースの管理ファイルが生成されることによって、例文を検索する時に、例文に関する情報を提供できるようにした。

現在、電子化された韓国語の翻訳例文がなかったので、市販されている翻訳書を入力して約8メガバイトの翻訳例文データベースを構築した。今後、類似例文検索

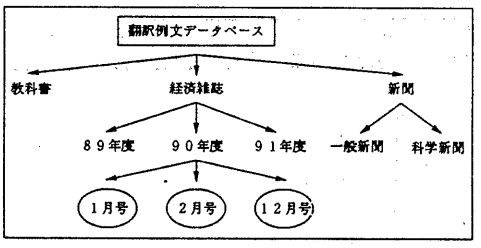
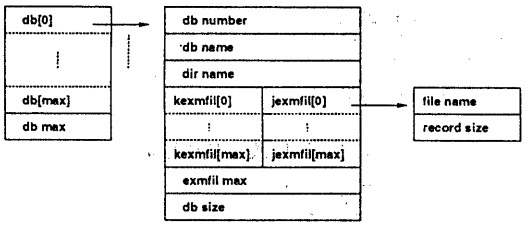



図 14: 翻訳例文データベースの階層構造




db : データベース

kexmfil : 韓国語例文ファイル　jexmfil : 日本語例文ファイル

図 15: 翻訳例文データベースの管理ファイル

の成功率を上げるためには、翻訳例文ファイルを増やす必要があると思われる。

## 6 実験および評価

### 6.1 実験

1. 文字照合検索法

図 16と図 17は文字照合検索法の典型的な例で、図 18は入力文字列にワイルド文字「*」を用いた例である。「*」とはどんな文字ともマッチするという意味である。

2. 意味照合検索法

図 19は意味照合検索の例で、文字照合検索の結果文より格パターンとともに意味的にも入力文に似ている翻訳例文が検索されていることが分かる。

### 6.2 評価

#### 6.2.1 検索時間の評価

1. 文字照合検索法

1 から 20 文字までの入力文の長さを 4 段階に分けて、各段階別に 50 文を対象にしてテストした。入力文字列の長さが大きくなるにしたがって、検索時間が増えていることが分かる。韓国語の検索時間は入力文字列の長さが 6 から 10 までのとき、最低 399 ミリ秒、最大 2,049 ミリ秒、平均 1,405 ミリ秒かかった。日本語の場合は最

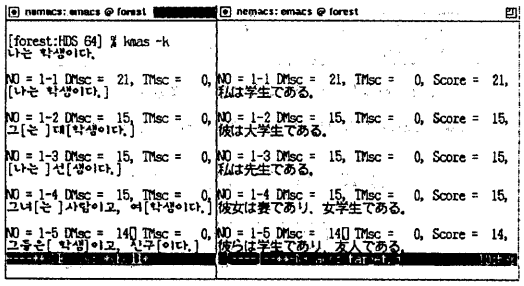



図 16: 文字照合検索例 - 韓日

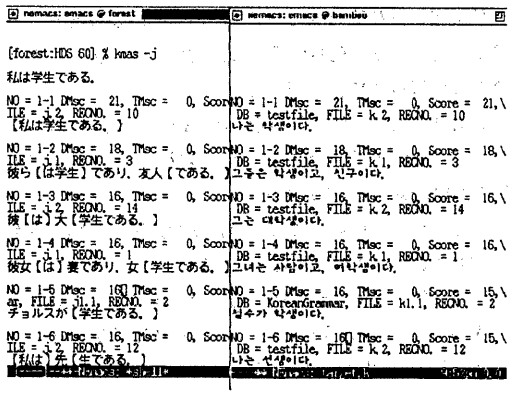



図 17: 文字照合検索例 - 日韓

低 516 ミリ秒、最大 2,299 ミリ秒、平均 1,543 ミリ秒かかった。[*5]表 20は韓国語の検索時間の評価結果を表している。

2. 意味照合検索法

1 から 20 文字までの入力文の長さを 2 段階に分けて、各段階別に 50 文を対象にし、分類番号照合の対象数を三つにしてテストした。このとき、「$」で囲まれている文字列は長さが 1 と数えた。検索時間の増減は主に分類番号照合の対象に対する処理にかかっているので、図 21から入力文字列の長さにはあまり変化がないということが分かる。韓国語の検索時間は入力文字列の長さが 5 から 10 までのとき、最低 2,916 ミリ秒、最大 6,499 ミリ秒、平均 4,026 ミリ秒かかり、日本語の場合は韓国語の検索時間の約 3 倍であった。

#### 6.2.2 検索結果文の評価

1. 文字照合検索法

1 から 20 文字までの入力文の長さを 4 段階に分けて、

[*5] 評価は SparcStation ELC で行なった。

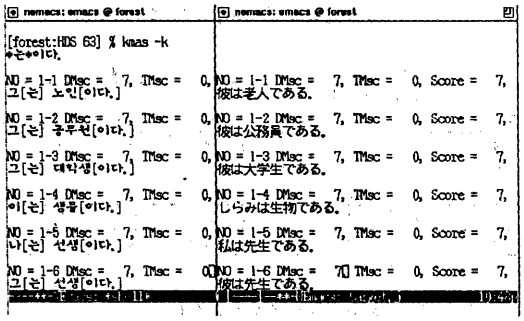



図 18: 文字照合検索のワイルド文字使用例 - 韓日

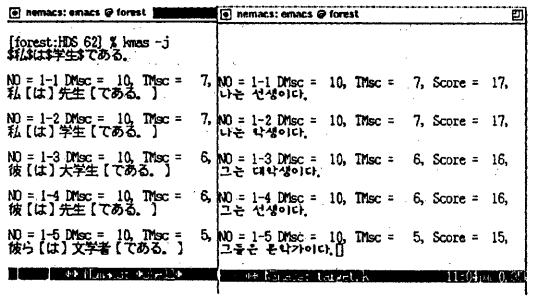



図 19: 意味照合検索例 - 日韓

各段階別に 50 文を対象にしてテストした。評価ランクは次のように四つのランクに分けた。

A 検索された例文が入力文に完全に一致する。

B 検索された例文が入力文の翻訳に充分な情報を与える。

C 検索された例文が入力文の翻訳に部分的な情報を与える。

F 検索された例文が入力文の翻訳にどんな情報も与えない。

そして、検索結果文の上位 5 文を取り出して、その中で一番良い結果文を選んで評価した。その評価結果は表 1と表 2のようである。入力文字列の長さが大きくなると、低いランクになっている。ところが、ランク C が与えている検索結果文も入力文の翻訳に部分的情報を提供しているので、これらを組み合わせると翻訳に役に立つと思われた。

2. 意味照合検索法

1 から 20 文字までの入力文の長さを 2 段階に分けて、各段階別に 50 文を対象にしてテストした。ここでは、分類番号照合の対象数が三つの場合を載せた。評価ランク

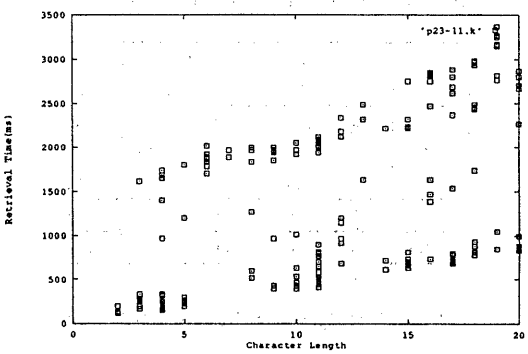



図 20: 文字照合検索法の検索時間の評価 - 韓日

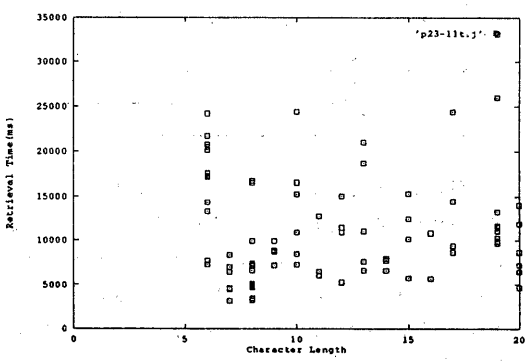



図 21: 意味照合検索法の検索時間の評価 - 日韓

は次のように四つのランクに分けた。

A 検索された例文が入力文に完全に一致する。

B 検索された例文が入力文のパターンと完全に一致しているが、意味的に似ていない分類番号照合対象の単語がある。

C 検索された例文が入力文のパターンと部分的に一致していて、意味的に似ていない分類番号照合対象の単語がある。

F 検索された例文が入力文のパターンと一致しなく、意味的に似ていない分類番号照合対象の単語がある。

そして、検索結果文の上位 5 文を取り出して、その中で一番良い結果文を選んで評価した。その評価結果は表 3のようである。

## 7 おわりに

用例を用いた翻訳支援システムには文字照合検索法と単語照合検索法が用いられている。韓国語と日本語は表

表 1: 文字照合検索法の検索結果文の評価 - 韓日

| Grade | Character Length 1–5 | 6–10 | 11–15 | 16–20 | Total |
|---|---|---|---|---|---|
| *A* | 29 | 5 | 4 | - | 38 |
| *B* | - | 11 | 15 | 13 | 39 |
| *C* | - | 28 | 15 | 29 | 72 |
| *F* | 21 | 6 | 16 | 8 | 51 |
| Total | 50 | 50 | 50 | 50 | 200 |

表 2: 文字照合検索法の検索結果文の評価 - 日韓 -

| Grade | Character Length 1–5 | 6–10 | 11–15 | 16–20 | Total |
|---|---|---|---|---|---|
| *A* | 21 | 9 | 4 | - | 34 |
| *B* | 4 | 18 | 4 | 3 | 29 |
| *C* | 3 | 20 | 29 | 35 | 87 |
| *F* | 22 | 3 | 13 | 12 | 50 |
| Total | 50 | 50 | 50 | 50 | 200 |

記に用いられる文字の種類が多く、類義語には同じ文字が使われている傾向が高いという特性を持っている。そして、我々は CTM で示された文字照合検索法が韓国語にも有効であろうと考えた。しかし、韓国語は日本語よりは文字種類が少なく、CTM の文字照合検索法の検索時間を改善する必要があると考えた。

そこで、本稿では CTM で用いられた文字検索照合法を改良し、韓国語に適用してみることによって、この方法が韓国語においても有効であることが分かった。まず、2 文字インデックステーブルと選抜基準値を用いることによって、より早く類似例文が検索できるようにした。しかし、文字照合検索法では、同じ文字が書かれていないと、類似例文として検索されない場合がある。そのため、文字照合検索法に分類語彙表の分類番号に基づいて意味的類似度を計算する分類番号照合法を加えた意味検索照合法を適用して、文字的には完全に一致しなくても意味的に一致している文であれば、類似例文として検索できるようにした。こうして、あらかじめ大量の翻訳例文データベースを構築しなければ類似例文として検索されなかった従来の例文検索システムの欠点が改善されたと思われる。

しかし、本システムでの検索時間と類似例文の検索成功率もまだ充分ではないと思われる。今後さらに検索時間を高速化するとともに、より多くの類似例文が検索できるように改良する必要があると思われる。

表 3: 意味照合検索法の検索結果文の評価

日韓

| Grade | Char. Length 1–10 | 11–20 | Total |
|---|---|---|---|
| *A* | 4 | 1 | 5 |
| *B* | 1 | 2 | 3 |
| *C* | 7 | 10 | 17 |
| *F* | 13 | 12 | 25 |
| Total | 50 | 50 | 100 |

韓日

| Grade | Char. Length 1–10 | 11–20 | Total |
|---|---|---|---|
| *A* | 8 | 1 | 9 |
| *B* | 4 | 3 | 7 |
| *C* | 9 | 11 | 20 |
| *F* | 29 | 10 | 39 |
| Total | 50 | 50 | 100 |

**参考文献**


[1] Nagao, M.: A Framework of a Mechanical Translation between Japanese and English by Analogy Principle, in ARTIFICIAL AND HUMAN INTELLIGENCE, Elsevier Science Publishers, pp173-180 (1984).

[2] Satoshi Sato: CTM:An Example-Based Translation Aid System Using the Character-Based Match Retrieval Method, Vol.4, pp.1259-1263, Proc. of COLING-92, Nantes, August 23-28 (1992).

[3] 隅田 英一郎、堤 豊: 翻訳支援のための類似用例の実用的検索法、電子情報通信学会論文誌、Vol.J74-D-II, No.10, pp.1437-1447 (1991).

[4] 中村 直人: 用例検索翻訳支援システム、情報処理学会第 38 回全国大会、4E-5 (1992).

[5] 国立国語研究所、分類語彙表、秀英出版 (1964).

[6] 黄 道三、長尾 真、佐藤 理史: 日本語の分類語彙表からの韓国語の分類語彙表の作成、情報処理学会自然言語処理研究会報告、94-12 (1993).

[7] 佐藤 理史: 用例検索による日英翻訳支援システム CTM2、研究報告書、北陸先端科学技術大学院大学 (1993).

[8] 黒橋 禎夫、長尾 真: 格フレーム選択における意味マーカと例文の有効性について、情報処理学会自然言語処理研究会報告、91-11 (1993).

[9] 長尾 真: 言語工学、昭晃堂 (1983).


**謝辞**


本研究で翻訳例文の収集に協力して頂いた長尾研の渡辺靖彦氏、韓国科学技術院の崔 紀善教授、韓国システム工学研究所の金 泰完氏に感謝します。