2F1-1

# 英語ウィキペディアを日本語で引く

Japanese-English Cross-Language Headword Search of Wikipedia

岡田昌也　佐藤理史　駒谷和範
Masaya Okada　Satoshi Sato　Kazunori Komatani

名古屋大学大学院工学研究科電子情報システム専攻
Department of Electrical Engineering and Computer Science, Graduate School of Engineering, Nagoya University

This paper describes a Japanese-English cross-language headword search system of Wikipedia, which enables to find an appropriate English article from a given Japanese query term. The key component of the system is a term translator, which selects an appropriate English headword among the set of headwords in English Wikipedia, based on the framework of non-productive machine translation (translation by selection). An experimental result shows that the translation performance of our system is equivalent to or slightly better than commercial machine translation systems, Google Translate and Mac-Transer.

## 1. はじめに

多くの人が、語 (ターム) の意味を調べるために、Web 上の百科事典『ウィキペディア』を利用する。ウィキペディアには多くの言語の版があるが、その中で規模が最大の英語ウィキペディア (EnWiki) は、日本語ウィキペディア (JaWiki) の約 5 倍の記事数を有する。

日本語話者にとっては、JaWiki に求める記事があれば、それが一番便利である。しかし、そのような記事がない場合、EnWiki の記事は次善の策となる。ある程度の英語力があれば、その英語記事から必要な情報を得ることができる。

ここで一つ、大きな問題が浮かび上がる。それは、「どうやって EnWiki を引くか」という問題である。いま、「結合性オービタル」について調べたい場合を想定しよう。JaWiki を引いても何も情報が得られない。では、EnWiki で調べようと、そのトップページを開いたのはいいが、いったい、どんなタームで引けばよいのだろうか。このような状況では、「結合性オービタル」に対応する英語タームは、当然、わからない。となると、EnWiki を引くことができない。

上記の問題は、EnWiki を日本語で引くことができれば解決できる。このように、別の言語で辞書や百科辞典を引くことを、言語横断見出し語検索 **(Cross-Language Headword Search: CLHS)** と呼ぼう。理想的には、「結合性オービタル」で JaWiki を引いたとき、該当する記事がなければ、そのまま EnWiki を引いてくれるシステムがあればよい。このようなシステムは、入力ターム (「結合性オービタル」) を適切に英訳することができれば実現できる。

ここでの翻訳は、EnWiki を引くための翻訳である。つまり、翻訳結果は EnWiki の見出し語となっていない限り意味がない。この制約を積極的に利用すると、「入力タームの英訳を EnWiki の見出し語の中から選ぶ」という考えが導かれる。我々がターム翻訳のための方式として提案した非生産型機械翻訳 **(Non-Productive Machine Translation)**[岡田 10] は、このような考え方に基づいている。

本論文では、我々が実装したウィキペディア用の日英 CLHS システムについて述べる。まず、2 節で作成したシステムの概要を述べ、3 節で本システムが採用した非生産型機械翻訳について説明する。4 節では、非生産型機械翻訳を日英ターム翻訳へ適用する拡張について説明し、5 節でテストセットを使用した本システムの評価について述べる。

連絡先: 岡田昌也，名古屋大学大学院工学研究科電子情報システム専攻，名古屋市千種区不老町 C3–1(631), masaya_o@nuee.nagoya-u.ac.jp

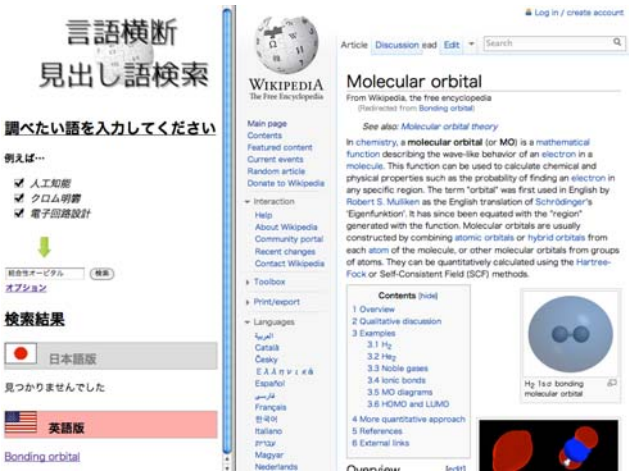


図 1: EnWiki を日本語で引くシステム

## 2. システムの概要

作成したシステムは、JaWiki および EnWiki への検索インタフェースとして動作する。本システムは、ユーザーとして日本語話者を想定しており、入力されたタームの記事が JaWiki に存在する場合はその記事を表示し、存在しない場合にのみ、EnWiki に対する CLHS を実行する。図 1 に、「結合性オービタル」に対する実行例を示す。JaWiki には「結合性オービタル」の記事は存在しないため、CLHS が実行され、得られた訳語 “Bonding orbital” を経由して、英語記事 “Molecular orbital” が表示される。

本システムは、次のように動作する。

1. 日本語ターム $s$ が入力される
2. **JaWiki** の見出し語を探す
   ターム $s$ が JaWiki の見出し語リストに存在するかどうかを調べる。存在する場合は、その記事を表示する。存

在しない場合は、ステップ 3 に進む。

3. **EnWiki に対して CLHS を実行する**
   EnWiki の見出し語リストの中から、$s$ の英訳を探す。得られた英訳の数によって、次の動作が異なる。
   (a) 英訳が **1** つだけ得られた場合
   得られた英訳で EnWiki を引き、記事を表示する。
   (b) 複数の英訳が得られた場合
   得られたすべての英訳を表示する。この後、いずれかの英訳をユーザーが選択すれば、その記事を表示する。
   (c) 英訳が **1** つも得られなかった場合
   英訳が見つからなかったことを表示する。

## 3. 非生産型機械翻訳

本システムの中核である CLHS の実現には、非生産型機械翻訳 [岡田 10] を用いる。非生産型機械翻訳は、タームの翻訳に特化した翻訳方式で、あらかじめ大規模な訳語候補リスト (ターゲットリスト) $T$ を準備しておき、この中から、翻訳すべきターム (ソースターム) $s$ の訳語を選択する。この方式では、翻訳のための知識源として、対訳辞書 $D$ (訳語対の集合) を利用する。例えば、日英対訳辞書では、$\langle$ 機械翻訳, machine translation$\rangle$ のような訳語対 (日本語の文字列と英語の単語列のペア) がその要素となる。

本方式では、訳語対の列 $\delta \in D^*$ が、より大きな訳語対、および、その部分対応を表現する。例えば、

$$\begin{aligned} d_1 &= \langle \text{機械翻訳}, \text{machine translation} \rangle \\ d_2 &= \langle \text{システム}, \text{system} \rangle \end{aligned}$$

とするとき、訳語対の列 $\delta = d_1 d_2$ は、

$$\begin{aligned} \delta &= d_1 d_2 \\ &= \langle \text{機械翻訳}, \text{machine translation} \rangle \langle \text{システム}, \text{system} \rangle \\ &= \langle \text{機械翻訳システム}, \text{machine translation system} \rangle \end{aligned}$$

という 3 つの情報を同時に表わす。なお、$\mathrm{src}(\delta)$ と $\mathrm{tgt}(\delta)$ は、それぞれ、訳語対のソース側、ターゲット側を表すものとする[*1]。

ソースターム $s$ が与えられると、本方式は、

$$\text{(ソース側条件)} \qquad \mathrm{src}(\delta) = s \tag{1}$$

$$\text{(ターゲット側条件)} \qquad \mathrm{tgt}(\delta) \in T \tag{2}$$

の 2 つの条件を満たす訳語対 $\delta \in D^*$ をすべて求め、そのターゲット側 $\mathrm{tgt}(\delta)$ を、$s$ の訳語として出力する。出力される訳語は 1 個とは限らず、複数の場合も、0 個の場合もある。

## 4. 日英ターム翻訳への適用

### 4.1 日英ターム翻訳に見られる現象

いま、日本語ターム $j$ から対応する英語ターム $e$ を求めることを考えよう。理論的には、(1) 対訳辞書 $D$ が訳語対 $\langle j, e \rangle$ を生成可能であり、かつ、(2) $e$ がターゲットリスト $T$ に含まれていれば、前節の方式で $j$ から $e$ を必ず得ることができる。しかしながら、現実には、次のような要因により、しばしば (1) が満たされない。

*1 これらの関数は、必要に応じて区切り記号 (スペース等) の挿入を行なうものとする

#### 4.1.1 日本語の表記ゆれ

日本語では、語の表記が複数存在する場合がある。タームの翻訳において特に顕著なのは、次の 3 種類の表記ゆれである。

1. 数字の表記ゆれ (漢数字と算用数字)
   「第２」と「第二」など
2. カタカナの表記ゆれ
   「モホロヴィッチ」と「モホロビッチ」など
3. 漢字の表記ゆれ
   「浸蝕」と「浸食」、「蛋白」と「タンパク」など

対訳辞書 $D$ の検索は、表記 (文字列) で行なうので、一方の表記のエントリーしか辞書に存在しない場合、もう一方の表記で検索した場合は、検索に失敗する (訳語を得ることができない)。

#### 4.1.2 トランスリタレーション

英日翻訳では、訳語に日本語を当てる以外に、カタカナ表記による音訳 (トランスリタレーション) を当てることがしばしば行なわれる。例えば、“orbital” は、通常「軌道」と訳されるが、「オービタル」と訳されることもある。しかしながら、対訳辞書には「軌道」という訳語しか記述されないことが多い。このような場合、「オービタル」の原綴を対訳辞書から得ることができない。

#### 4.1.3 付属語・接尾辞の消失・出現

専門用語の多くは、英語から日本語へ翻訳されたものである。この過程で、ある種の変形が導入されることがある。例えば、“multifactorial inheritance” は、各単語の部分訳をつなげれば「多因子の遺伝」となるが、日本語訳には「の」を削除した「多因子遺伝」が当てられる。これとは逆に、“bonding orbital” は「結合オービタル」ではなく、「性」が挿入されて「結合性オービタル」となる。このように、複合語の英日翻訳においては、ある種の付属語や接尾辞が削除されたり挿入されたりする。その逆翻訳である日英翻訳でも、これらの現象に適切に対処しなければ、正しい訳語が得られない。

#### 4.1.4 語順の変更

複合語の英日翻訳では、ほとんどの場合、語の順序は保持される。しかしながら、“of” を含む複合語は、その例外となる。例えば、“acceleration of free fall” は、「自由落下の加速度」と訳される。3 節で述べた翻訳方式は、このような語順変更に対応していない ($D^*$ には、語順を変更する訳語対は含まれない)。

### 4.2 日英ターム翻訳のための拡張

上記の問題に解決するために、次のような拡張機構を導入する。

#### 4.2.1 表記ゆれを考慮した辞書引き

日本語の表記ゆれは、対訳辞書に実質的にはエントリーが存在するのに、表記ゆれによる文字列の不一致のため、そのエントリーが検索できない事態を引き起こす。この問題は、辞書引きの段階で、典型的な表記ゆれに対処することで軽減できる。

数字の表記ゆれには、漢数字と算用数字の変換規則を導入すればよい。同様に、カタカナの表記ゆれも、以下の変換規則で処理する。

$$\{\ \text{ヴィ} \leftrightarrow \text{ビ},\ \text{ヴァ} \leftrightarrow \text{ワ},\ \text{ヴァ} \leftrightarrow \text{バ},\ \text{ッ} \leftrightarrow \epsilon,\ \text{ー} \leftrightarrow \epsilon \}$$

ここで、$\epsilon$ は、空文字を表し、「ッ」や「ー」は削除されうることを意味する。いずれの変換規則も、辞書引きの対象となる日本語文字列に数字やカタカナが含まれる場合に適用する。

これらの規則は、仮想的に、「第２ (second)」に対して「第二 (second)」を、「コンピューター (computer)」に対して「コンピュータ (computer)」を辞書に追加したことに相当する[*2]。

漢字の表記ゆれは、読みで辞書を引くことで対処する。このために、まず、対訳辞書の全てのエントリーの日本語側に形態素解析器を用いて読みを付与する。同様に、辞書引きの対象となる日本語文字列の読みを形態素解析器を用いて求め、読みの一致するエントリーをすべて検索する。この方法により、「浸蝕 (しんしょく)」から、「浸食 (しんしょく; erosion)」を得ることができる。しかしながら、この方法は、読みが同じものをすべて検索するため、誤った訳語を得てしまう可能性も高い。

#### 4.2.2 トランスリタレータの導入

対訳辞書に含まれないトランスリターレションに対処するために、トランスリタレータを導入する。具体的には、辞書引きの際に、対象となる日本語文字列がカタカナ文字列の場合は、対訳辞書検索に加え、トランスリタレータを実行し、得られた原綴候補も訳語として採用する。これも、一種の対訳辞書拡張とみなすことができる。

#### 4.2.3 付属語・接尾辞を考慮した辞書引き

複合語の英日翻訳における付属語・接尾辞の消失・出現も、辞書引きの拡張により対処できる。付属語・接尾辞の消失は、例えば、「多因子」に対する訳語を、末尾に「の」をつけた「多因子の」でも検索することで解決できる。このような付属語・接尾辞には、次のものがある。

{ の, 的, する, 的な, な, 性の, 性, 用, 法, 論, 式, 化, 型, 術, 症, 病 }

一方、付属語・接尾辞の出現は、例えば、「結合性」に対する訳語を、末尾の「性」を削除した「結合」でも検索することで解決できる。このような付属語・接尾辞には、次のものがある。

{ 性, 症, の, 術, 法, 型, 的, 化, 類, 用, 論, 式, 病 }

#### 4.2.4 ターゲットリストの拡張—語順の変更—

語順の変更に対処するための拡張は、いささかトリッキーである。ここでは、「自由落下の加速度 (acceleration of free fall)」を例として説明する。まず、“of” を含む英語ターム (“A of B”) に対して、“of” を含まない “B A” なる形を生成してターゲットリスト $T$ に加えると同時に、元の “A of B” へのリンクを張っておく。このような準備の後、「自由落下の加速度」を翻訳すると、「free fall (自由落下の) acceleration (加速度)」が得られ[*3]、その後、リンクをたどることにより、“acceleration of free fall” が得られる。

### 4.3 拡張の優先順位

上記の拡張機構の導入は、対訳辞書 $D$ の拡大、あるいは、ターゲットリスト $T$ の拡大を意味する。すなわち、拡張機構の導入により、出力される訳語の数は増加する。その一方で、誤った訳語を出力する危険性も高くなる。そのため、比較的安全な拡張機構から優先して使用し、訳語が一つも得られない場合のみ、より危険な拡張機構を使用する。表 1 に拡張レベルとそれぞれのレベルで使用する拡張機能を示す。

## 5. 実験

本節では、3 節と 4 節で述べた方法を、テストセットを用いて評価した結果について述べる。

*2 「コンピュタ (computer)」も追加されてしまうが、そのような文字列が入力されなければ実害はない。

*3 「の」は付属語・接尾辞を考慮した辞書引きで削除される。

表 1: 拡張レベル一覧

| | $L_0$ | $L_1$ | $L_2$ |
|---|---|---|---|
| 数字表記ゆれ | √ | √ | √ |
| カタカナ表記ゆれ | √ | √ | √ |
| トランスリタレーション | √ | √ | √ |
| 付属語・接尾辞の消失 | | √ | √ |
| 付属語・接尾辞の出現 | | √ | √ |
| 漢字表記ゆれ | | | √ |
| 語順変更 | | | √ |

表 2: 各条件を満たすテストペア $\langle j, e\rangle$ の数（9,542 ペア中）

| 集合 | $\langle j, e\rangle$ が満たす条件 | 数 |
|---|---|---|
| $H$ | $j$ は、Web 上に 10 回以上出現する | 8,672 (90.9%) |
| $\overline{W_j}$ | $j$ は、JaWiki の見出し語ではない | 3,914 (41.0%) |
| $W_e$ | $e$ は、EnWiki の見出し語である | 8,344 (87.4%) |
| $T$ | $H \cap \overline{W_j} \cap W_e$ | 2,549 (26.7%) |

### 5.1 使用したデータおよびツール

実験では、次のデータおよびツールを使用した。

1. 対訳辞書 $D$
   英和辞書『英辞郎』ver.116 [*4] から作成した。具体的には、『英辞郎』に収録されている訳語対と、それらから抽出した部分訳語対 ([外池 07], [藤井 00]) を併せたものを用いた。そのサイズは、2,366,870 ペア (日本語: 1,507,143; 英語: 1,887,892) である。
2. ターゲットリスト $T$
   英語ウィキペディア[*5] の見出し語リスト (5,907,150 件) を用いた。
3. トランスリタレータ
   『絣』[Sato 10] を用いた。
4. 読みを求めるための形態素解析器
   MeCab[*6] + UniDic [*7] を用いた。

### 5.2 テストセット

テストセットは、『オックスフォード科学辞典』[Daintith 09] から作成した。まず、この辞典の索引から訳語対 9,542 ペア (日本語: 8,986; 英語: 8,871) を収集した。次に、テストセット $T$ を次のように定めた。

$$T \quad = \quad H \cap \overline{W_j} \cap W_e$$

この式に現れる 3 つの部分集合 $H$、$\overline{W_j}$、$W_e$ の意味と大きさを表 2 に示す。本実験では、(1) 実際に入力される可能性が高く (=$H$)、(2) JaWiki の記事が存在せず (=$\overline{W_j}$)、(3) EnWiki の記事が存在する (=$W_e$)、2,549 ペアをテストセットとして用いた。

### 5.3 実験結果と検討

テストセットに含まれるそれぞれのペア $\langle j, e\rangle$ に対し、その日本語側 $j$ を入力したとき、どのような出力が得られるかを調べた。出力されたそれぞれの訳語 $e'$ は、EnWiki において、$e$

*4 http://www.eijiro.jp/

*5 http://en.wikipedia.org/wiki/

*6 http://mecab.sourceforge.net/

*7 http://www.tokuteicorpus.jp/dist/

表 3: 実験結果 (1): 英辞郎との比較

| | 英辞郎 | 英辞郎 w/$T$ | 本方式 |
|---|---|---|---|
| perfect | 633 (24.8%) | 977 (38.3%) | 1,257 (49.3%) |
| ambiguous | 530 (20.8%) | 186 ( 7.3%) | 526 (20.6%) |
| subtotal | 1,163 (45.6%) | 1,163 (45.6%) | 1,783 (70.0%) |
| false | 346 (13.6%) | 148 ( 5.8%) | 357 (14.0%) |
| none | 1,040 (40.8%) | 1,238 (48.6%) | 409 (16.0%) |

と $e'$ が同一記事を指し示す場合に正解と判定した*8。それぞれの入力 $j$ に対する判定は、本システムの動作 (2 節) を考慮して、次の 4 種類に分類した。

a. Perfect
正解訳のみを出力 (= 正解記事を表示)
b. Ambiguous
正解訳の他に、他の訳語 (不正解訳) を出力 (= 表示される訳語リストに正解が含まれる)
c. False
不正解訳のみを出力 (= 不正解記事を表示、あるいは、表示される訳語リストに正解が含まれない)
d. None
出力なし

表 3 に実験結果を示す。この表では、ベースラインとして、翻訳に『英辞郎』単体を用いた場合、および、『英辞郎』とターゲットリスト $T$ による訳語限定を組み合わせた場合を、併せて示した。なお、この subtotal (=perfect+ambiguous) の数は、正解訳語を出力できた入力数を示す。本方式の subtotal の数は、ベースラインの 45.6%から 70.0%に大幅に増加している。このことは、タームの翻訳が、対訳辞書をそのまま引くだけでは不十分であり、ある種の合成操作が不可欠であることを示している。ターゲットリスト $T$ による訳語限定は、subtotal の数の増加には寄与しない。しかしながら、候補の数を抑制し、perfect の割合を高める効果がある。単純に複数の要素を合成すると、合成される訳語の数はそれぞれの要素の訳語候補数の積となるが、本方式では、ターゲットリスト $T$ の効果により、ambiguous の場合の平均訳語出力数は 3.6 個であり、十分少ない数に抑えられている*9。そのため、得られた訳語リストをそのままユーザーに提示しても、ユーザーが困惑することはない。以上をまとめると、本方式は、全体の 70%の入力に対して、該当する EnWiki の記事へのアクセスを可能にするといえる。なお、本方式の実行速度は平均 0.16 秒であり、実用に十分である。

本方式の性能を他の翻訳システムと比較した結果を表 4 に示す。ここでは、『Google 翻訳』*10、および、市販の翻訳ソフト『MAC-Transer 2010』*11 と比較した。これら 2 つの翻訳システムは、1 件の入力に対し、高々1 件の訳語しか出力しない。これらのシステムと公平な評価を行なうために、本方式が訳語を 2 件以上出力した場合は、サーチエンジンを利用して入力と各訳語とのアンドヒット数を求め、この数が最も多かったものを最終的な出力とする後処理を追加した。

*8 Wikipedia には、リダイレクトと呼ばれる機構があり、複数の同義語に対して同一の記事が表示される。
*9 false の場合の平均訳語出力数は 3.0 個である。
*10 http://translate.google.co.jp/#, 2011/5/1 現在
*11 http://www.crosslanguage.co.jp/products/mac2010/

表 4: 実験結果 (2): 翻訳システムの比較

| | Google | Transer | 本方式 |
|---|---|---|---|
| perfect | 1,530 (60.0%) | 1,591 (62.4%) | 1,648 (64.7%) |
| false | 977 (38.3%) | 892 (35.0%) | 492 (19.3%) |
| none | 42 ( 1.6%) | 66 ( 2.6%) | 409 (16.0%) |

表 4 において、perfect の数は、本方式が若干多いが、それほど大きな差がない。Google 翻訳で利用される膨大なデータ量、および、Transer の開発に費やされた労力と比較すると、本方式は、タームの翻訳という限定された用途において、非常に簡便な方法で同程度の性能を得ることができる。これは、本方式の利点である。なお、本方式が他の 2 つのシステムに比べ、false の件数が少なく、none の件数が多いのは、ターゲットリスト $T$ による訳語限定の結果である。

本実験では、$e \in T$ なる条件を満たすテストペア $\langle j, e\rangle$ を用いた。このため、正解訳 $e$ を出力できない原因は、(拡張された) 対訳辞書 $D$ が訳語対 $\langle j, e\rangle$ を生成できないことによる。さらに原因を遡れば、次のいずれかの原因に行きつく。

1. 訳語対が要素から合成できない
例：⟨ 保水量, field capacity⟩
2. 要素合成に必要な部分訳語対が存在しない
例：⟨顕示 行動, display behavior⟩
3. “A of B” 以外で、語順が変更される
例：⟨ 五臭化リン, phosphorus pentabromide⟩

要素から合成できない訳語対 (上記の 1) は、それ全体を対訳辞書に登録しておく必要がある。つまり、語順が変更される若干の例外 (上記の 3) を除けば、結局のところ、対訳辞書 $D$ の網羅性と品質がシステムの能力を規定する。これは、機械翻訳システムの宿命であり、本方式もその宿命からは逃れられない。しかしながら、本方式では、ターゲットリスト $T$ を用いて訳語を制限するので、対訳辞書を拡大利用しても、誤訳の生成が適度に抑制される。今後は、対訳辞書 $D$ の網羅性向上を目指し、シソーラスを利用した辞書拡張を検討する予定である。

謝辞 本研究は、科学研究費補助金挑戦的萌芽研究 (課題番号 22650047) の支援を受けている。

## 参考文献

[Daintith 09] Daintith, J. ed., 山崎 昶 (訳): オックスフォード科学辞典, 朝倉書店 (2009)

[Sato 10] Sato, S.: Non-productive machine transliteration, in *Adaptivity, Personalization and Fusion of Heterogeneous Information*, RIAO '10, pp. 16–19 (2010)

[岡田 10] 岡田 昌也, 佐藤 理史：大規模訳語候補集合を利用した専門用語翻訳, 第 24 回人工知能学会全国大会論文集, 2C4-1 (2010)

[外池 07] 外池 昌嗣, 宇津呂 武仁, 佐藤 理史：ウェブから収集した専門分野コーパスと要素合成法を用いた専門用語訳語推定, 自然言語処理, Vol. 14, No. 2, pp. 33–68 (2007)

[藤井 00] 藤井 敦, 石川 徹也：技術文書を対象とした言語横断検索のための複合語翻訳, 情報処理学会論文誌, Vol. 41, No. 4, pp. 1038–1045 (2000)